Call-A.P.P.L.E.™

Volume VI, Number 9 September 1983 $2.50 ($3.00 Canada, £1.75 UK.)

COMPUTER FUTURES:
The Age of Silicon
Page 7

D IS FOR DATA:
Where Has the Data Gone
Page 11

AN APPLE FOR THE TEACHER:
BASIC vs LOGO
Page 15

Graphics Feature Section
A Call —A.P.P.L.E. LANGUAGE REVIEW:
Ceemac
Page 21

Apple PugetSound Program Library Exchange

TRUNKS FOR THE MEMORIES.

Introducing the most logical place to store Elephant Memory Systems® (or lesser brands of disks): The Trunk.

With its alphabetized library index, you can file or retrieve up to 60 disks, instantly.

The Trunk is made of durable molded plastic with a hinged, one-piece lid, to keep disks safe from dust, dirt, and other detriments which disks despise.

And, it's portable. Because the lid doubles as a carrying handle so your Elephant Memory Systems® disks can go anywhere you do.

There's a model for 5¼″ and 8″ floppies, as well as a cassette-and-game file and a special Atari® version.

So if you're looking for the best disk storage system on the market . . .

The Trunk is an open-and-shut case.

THE TRUNK. ENDORSED BY ELEPHANTS.

Elephant Memory Systems® Disks

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a life-time of heavy-duty use).

Marketed exclusively by Leading Edge Products, Information Systems and Supplies Division, 55 Providence Highway, Norwood, MA 02062. Dealers: Call toll-free 1-800-343-8413; or in Massachusetts, call collect (617) 769-8150.

Circle 40 on Reader Service Card

Predictable.

A flawless future is in sight with 3M diskettes.

When it comes to keeping track of precious data, predictable means reliable. Being able to count on every diskette, every time. At 3M, reliability is built into every diskette. We've been in the computer media business for over 30 years. And we've never settled in. We're constantly improving and perfecting our product line, from computer tape and data cartridges to floppy disks.

3M diskettes are made at 3M. That way, we have complete control over the entire manufacturing process. And you can have complete confidence in the reliability of every 3M diskette you buy.

Look in the Yellow Pages under Computer Supplies and Parts for the 3M distributor nearest you. In Canada, write 3M Canada, Inc., London, Ontario. If it's worth remembering, it's worth 3M diskettes.

3M hears you...

3M

Circle 39 on Reader Service Card

Apple PugetSound Program Library Exchange

GENERAL OFFICES
21246 - 68th Ave. S.
Kent, WA 98032 - (206) 872-2245
Hours: 9am to 4pm PST, Mon-Fri.

Officers and Directors

Dick Hubert	President
Bob Huelsdonk	Vice-President
Dave Lingwood	Secretary
Bob Clardy	Treasurer
Val J. Golding	Director
Ken Smith	Director
Don Williams	Director

Administration

Dick Hubert	Executive Director
Don Steele	Asst. to Executive Director
Dave O'Leary	Controller
Sally Davies	Manager of Operations
Steve Johnson	Technical Staff Manager
Charles Stillman	Manager of Education Services
Kevin Donohoe	Manager of Marketing

Departments

Order desk
Tracing
Member Information
Education, Classes
Retail store
Dealer services
Marketing
Technical support
Testing and Evaluation
Operations

Editorial Department

5606 California Ave. SW
Seattle, WA 98136 - (206) 932-8288

Val J. Golding	Editor
Kathryn Hallgrimson Suther	Associate Editor
Bruce van Norman	Pascal Editor
Grawin Publications	Advertising
Jonathan Post	Contributing Editor

Staff writers

Cornelis Bongers
Jim Hockenhull
Clifton M. Howard, M.D.
Ken Kashmarek
Bill Parker
Lee Reynolds
Clay Ruth
Michael Shadick
David G. Sparks
Rick Sutcliffe
Ralph H. Swerdlow, M.D.

Member Services

24 hour toll-free order line:
1-800-426-3667
(except Washington state)
Orders ONLY
Information not available

Programming/Technical Assistance
Call -A.P.P.L.E. HOTLINE
(206) 872-9004
9am to 3pm - 6pm to 10pm
(Pacific time)
7 days per week

A.P.P.L.E. Crate Bulletin Board
(206) 935-9119

Membership Information
is available on postpaid tear-off card inside.

Meetings
7:30pm third Tuesday of each month
A.P.P.L.E. Office
21246 68th Ave. S., Kent
(except as noted)

A.P.P.L.E. on the SOURCE

A special section of The Source is open only to A.P.P.L.E. members who may download Call -A.P.P.L.E. programs. To access, type:

CALL-APPLE from Source command mode. To register, send mail to ST1570 with your name, membership number, expiration date and Source account number.

Prospective authors please send a large self-addressed stamped envelope (37¢ postage) to Editorial Dept. for a copy of Author's guidelines.

Advertising inquiries please contact:
GRAWIN PUBLICATIONS
1020 Lloyd Building, Seattle, WA 98101
(206) 223-0861

Call —A.P.P.L.E.

MONTHLY MEMBER SPECIAL

THE Spreadsheet

Super Memory Special
Offer good till 11/1/83

Product code: STSS-HP28

Check the value!

√ 128K RAM card

√ THE Spreadsheet version 2.0

Both for: **$325**

Regular price: $370
Equivalent retail value: $645

Sorry, not available overseas.

24 hour toll-free order no.
800-426-3667
(except in Washington)
Washington state residents add 7.9% sales tax.

A.P.P.L.E. Orders
21246 68th Ave. S.
Kent, WA 98032
(206) 879-2245

AD INDEX

A.P.P.L.E.	2, 77
Access Micro	83
American Data Cable	83
Anthro-Digital	76
Apple Computer Inc.	48, 49
Avant-Garde	52,53
B & D Electronics	86
Beagle Brothers	6
Bytes & Pieces	27
Cases, Inc.	10
Computer Case	59
Computer Software Associates	87
Control Data Corp.	IBC
Dark Star	54
David-Data	24
Douglas Electronics	32
Gourmet Delicious Software	63
Key Tech	86
Laing Electronics	28, 76
Leading Edge Products	IFC, BC
LJK, Inc.	89
Micro Analyst	60
Micro Max	67
MicroPro International	38
Microware Distributing	74
Northeast Expositions	20
Nibble	92
Nibble Notch	87
Orange Micro	44, 45
Orange Plus Computers	64
Orbital Systems	42
The Professor	72
Softkey Publishing	31
Softraders International	24
Softronics	50
Software Banc	83
Software Masters	28
Southwestern Data Systems	36
Telephone Software Connection	67
3M Data Recording Products	1
Transtar	4
Utilico Software	10
VF Associates	72, 74
Zoom Telephonics	56

REQUEST FOR CHANGE OF ADDRESS

Dear A.P.P.L.E.,
I am moving and my present address is:

PRESENT:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

EFFECTIVE DATE _____

NEW ADDRESS:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

MEMBERSHIP NUMBER _____

* Please fill out and return this notice six weeks before moving.

Call-A.P.P.L.E.™

Volume VI, Number 9 September 1983/Menu

Val J. Golding, Editor
Kathryn Hallgrimson Suther, Associate Editor

ENTRÉES

7 COMPUTER FUTURES:
The Age of Silicon
Jonathan Post/contributing editor

11 D IS FOR DATA:
Where Has the Data Gone
Ken Kashmarek/Call —A.P.P.L.E. staff writer

15 AN APPLE FOR THE TEACHER:
BASIC vs LOGO
Lee Reynolds/Call—A.P.P.L.E. staff writer

21 **Graphics Feature Section**
A Call —A.P.P.L.E. LANGUAGE REVIEW:
Ceemac
Jim Hockenhull/Call —A.P.P.L.E. staff writer

25 A Call —A.P.P.L.E. SOFTWARE REVIEW:
The Graphics Magician
Harvey Gratt

29 **Applesoft Graphics Mover**
Homer O. Porter

33 **Shape Table Splicer**
Cyrus W. Roton

A LA CARTE

37 **Extra Memory for The Spreadsheet**
George Mochizuki

39 AN APPLE FOR THE TEACHER:
Grade Book Review
Kathryn Suther/Wes Felty/Pat Buck

43 **A.P.P.L.E. Computer Camp '83**
Charlie/Susan Stillman

47 A Call —A.P.P.L.E. SOFTWARE REVIEW:
Rocky's Boots
Earle Hancock

51 **Magic Square Dance**
J. Taylor

55 THE NORTHERN SPY:
Nellie and the Pirates
Rick Sutcliffe/Call —A.P.P.L.E. staff writer

61 A Call —A.P.P.L.E. TECHNOTE:
Ignoring ?Extra Ignored
Ari/Dov Freund

68 A Call —A.P.P.L.E. TUTORIAL:
Preparing A Menu
Val J. Golding

69 CP/MAGIC:
A Softcard LST Driver
Gerald Cahill

73 FROM THE HEART OF THE MONITOR:
And the Winner Is...
Ken Kashmarek/Call —A.P.P.L.E. staff writer

75 **SYSTEM.LIBRARY**
David Walker

77 **Pascal 1.2**
Allen Munro

78 **ApMail Apple-cations**
Ken Kashmarek/Call —A.P.P.L.E. staff writer

80 **The Gaming Room**
Val J. Golding

81 **Inverse String Printer**
Stowe Keller

82 **How to Become An Author**
Daniel T. Lyons

84 **Telecommunications with Filedump**
Steve Ross/Val J. Golding

87 ELECTRONIC MAIL:
Another Use for the Apple
Bruce Bourbonnais

HORS d'OEUVRES

5 The Editor Bytes Back
the editor

5 Boot
Craig MacIntosh

65 A.P.P.L.E. Doctor
Bruce Field

10, 52, 76, 95, 96 So What Did You Expect?

89 Write —A.P.P.L.E.
the readers

91 Tomorrow's Apples Today
Kevin O. Donohoe/editor

95 Call —A.P.P.L.E. Consultants

95 A.P.P.L.E. Order Form

96 Call —A.P.P.L.E. Unclassifieds

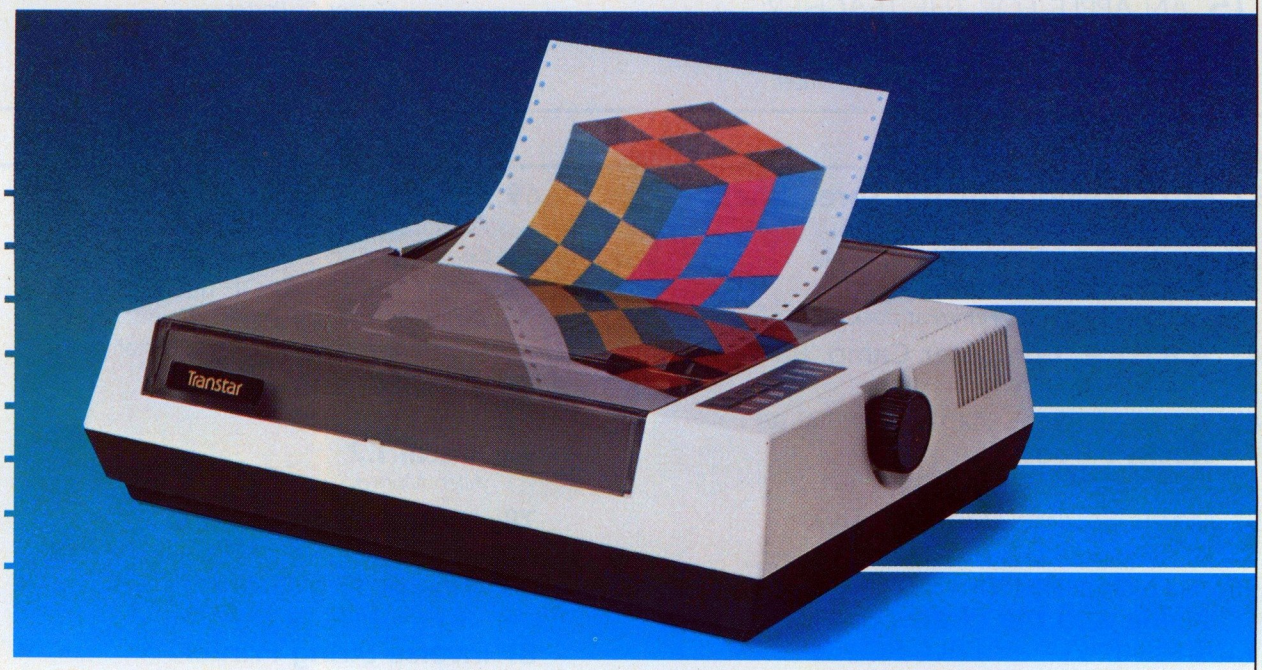
Apple user groups may reprint without prior permission any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commerical applications of programs herein, the author should be contacted c/o General Office. Call —A.P.P.L.E. is published monthly by Apple Puget Sound Program Library Exchange, 21246 - 68th Ave. S., Kent, WA 98032. Subscriptions are not available but copies are mailed free as a benefit of A.P.P.L.E. membership.

Entire contents copyright© 1983 by

Apple PugetSound Program Library Exchange

Innovative technology brings you quality color printing

for only $599



The technology of the all-new Transtar 315 color printer revolutionizes color impact printing! Unlike old-fashioned printers needing multiple passes with pins to print color, the Transtar 315 employs an innovative 4-hammer print head to allow 7 colors and more than 30 shades to be printed in a **single pass!**

A unique 4-color diagonal ribbon maximizes the efficiency of the 315's color imaging and enhances its simple reliability. Built by Seikosha, the most experienced manufacturing company of the famous Seiko group, the Transtar 315 is **available now** and has been designed to be compatible with the IBM, Apple II/IIe and Franklin personal computers. An optional PICS card also allows Apple and Franklin users to simply depress the 315's "copy" button to print any high- or low-resolution screen without exiting a program!

Transtar is bringing the technology of tomorrow to you today. **Your future in color printing is only $599 away.**

Transtar

P.O. Box C-96975, Bellevue, Washington 98009

Circle 41 on Reader Service Card

THE EDITOR BYTES BACK

the editor

SON-OF-A-GUN! It sure is interesting when we look back over some of the past issues of **Call -A.P.P.L.E.** to see where we're coming from, as we approach (just three months away) our seventh year of continuous publication. The profile of the typical Apple user has changed much over that period. The dyed-in-the-wool programmer is still there in great numbers, along with the tinkerer and the hobbiest, and you can bet just about every one of them, many now leaders in the computer industry, read **Call -A.P.P.L.E.,** but as it turns out, the Apple][has gone far beyond the sophisticated game and hobby machine that **Woz** in his wisdom, dreamt.

We have seen VisiCalc developed for the Apple, along with a multitude of clones and enhanced clones (such as our **THE Spreadsheet,** data bases and myriads of business and educational applications programs defying description in a single sentence. Many of these programs could have only been developed on the Apple because of its extreme versatility and because Apple Computer, Inc. (thank you again, **Woz**) saw fit to produce it as an "open" machine, i.e., with its "guts" bared for one and all to tinker with, and through the information published in **Call -A.P.P.L.E.** Incomplete results from our June readers choice card show **Steve Ross'** review of Spreadsheets a clear winner, indicative of our readers' current interests.

We hope you have noticed, bit by bit (or byte by byte), the additional material that has been appearing in our pages and the introduction of new columns such as **CP/Magic, An Apple For the Teacher, The Northern Spy, What in the Blaise** and **The Gaming Room.** On the road, but not here yet is **The 80-Column Column,** a business feature, a full-blown series of spreadsheet applications and still more to follow.

Two of our lead articles this month represent another innovative step: the continued story. Contributing editor **Jonathan Post** appears with his first major story, **Computer Futures,** excerpts from his book under way. Part 1, **The Age of Silicon,** serves as an introduction to the series which will include portions on robotics (look out, **Mike Thyng),** artificial intelligence, exotic techniques and more. But better yet, Jonathan is going to give you the reader a chance to participate interactively in his book! If you can't stand the suspense, turn to page 7 right now.

Prolific staff writer **Ken Kashmarek** introduces a four-parter, **D is For Data,** a treatise on Personal File Manager (PFM) with some descriptions of its file structure and how its files can be read with standard DOS. This feature will continue for the balance of the year, and it's one you'll not want to miss.

Educators note: our third cover item, **BASIC vs LOGO** was selected to help you understand the LOGO language and how it relates to BASIC. Staffer **Lee Reynolds** offers many examples and explanations on how to take a BASIC statement and write an equivalent in LOGO. Here's your opportunity to learn a second language.

And talking about languages, our **Graphics Feature Section** is headed by staff writer **Jim Hockenhull's** review of **Ceemac.** Ceemac is a visual composition tool, a graphics programming language, and his in-depth coverage may prompt you to give it a try. Also included in our graphics section is **Harvey Gratt's** review of Penguin's **The Graphics Magician.**

Some time ago **Steve Ross** put together a modem utility for our in-house use which utilized the popular **Filedump** patch to eliminate the need for a terminal program to transmit to another machine or system, then we added a few embellishments to produce **Teledump.**

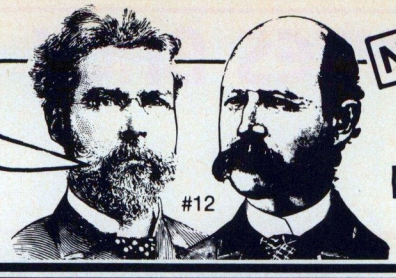
(continued on page 47)

BOOT

Craig MacIntosh

• 1982 Craig MacIntosh

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE, AND COMPATIBLE WITH APPLE II, II+ AND IIe.*
(Don't Settle for Less!)



*APPLE is a Registered Trade Mark of You-Know-Who.

APPLE MECHANIC

SHAPE-WRITER / BYTE-ZAP UTILITY
by BERT KERSEY

SHAPE EDITOR: Keyboard-draw shapes for animation in your programs. Create **Proportionally-Spaced Typefaces** with special characters. Six fonts included. Listable Applesoft demos show you how to animate graphics and create professional Charts and Graphs.

BYTE-ZAP: Rewrite any byte on a disk for repair or alteration. Load entire sectors on the screen for inspection. **Hex/Dec/Ascii** displays and input. Educational instructions include experiments for making trick file names, restoring deleted files, changing DOS, etc.

MORE: Useful music, text and hi-res tricks for your programs. Clear educational documentation.

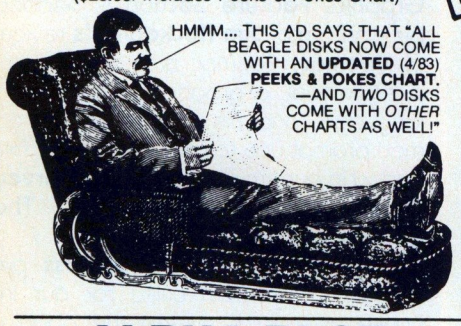
(**$29.50**: Includes Peeks/Pokes Chart & Tip Book #5)

APPLE MECHANIC TYPEFACES

26 NEW FONTS for use with Apple Mechanic programs. Many sizes of fully-editable characters.

BEAGLE MENU: Display only the file names you want from your disks (for example, only *Applesoft* files or only *Locked* files) for fast one-key cursor selection.

(**$20.00**: Includes Peeks & Pokes Chart)



ALPHA PLOT

HI-RES GRAPHICS/TEXT UTILITY
by BERT KERSEY and JACK CASSIDY

DRAW IN HI-RES on both pages using keyboard OR paddles/joystick. View lines before plotting. Use mixed-colors and Reverse (background opposite) plotting. Fast hi-res circles, boxes and ellipses; filled or outlined.

Compress Hi-Res Images to 1/3 Disk-Space. Superimpose hi-res pages 1 and 2 or re-locate any rectangular image area anywhere on either hi-res page.

Proportionally-Spaced Hi-Res Type with adjustable character size and color. Upper and lower case with no htab/vtab limits. Sideways typing for graphs too.

(**$39.50**: Includes Peeks/Pokes Chart & Tip Book #4)

DOS BOSS

DISK COMMAND EDITOR
by BERT KERSEY and JACK CASSIDY

RENAME DOS COMMANDS and Error Messages. "Catalog" can be "Cat"; "Syntax Error" can be "Oops" or *anything*. **Protect your programs**—An unauthorized Save-attempt can produce a "Not Copyable" message, or *any* message you want. Also List-Prevention and one-key program-execution from catalog.

Customize DOS. Change the Disk Volume heading to your message or title. Omit or alter catalog file codes. Fascinating documentation, tips and experiments.

ANYONE using your disks (booted or not) will be formatting DOS the way YOU designed it.

(**$24.00**: Includes Peeks/Pokes Chart & Tip Book #2)

TIP DISK #1

by BERT KERSEY

100 LISTABLE PROGRAMS from Beagle Bros Tip Books 1-4. Make your Apple do things it's never done! All 100 programs are changeable for experimentation. Includes Apple COMMAND CHART with all Applesoft, Integer Basic & DOS Commands and Descriptions!

(**$20.00**: With 2 Charts: Peeks/Pokes & Commands)

Beagle Bros
Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

UTILITY CITY

21 USEFUL UTILITIES by BERT KERSEY

LIST FORMATTER prints each program statement on a new line. Loops indented with printer Page Breaks. A great Applesoft program de-bugger! Also...

Multi-Column Catalogs for printouts, auto-post Run-number & Date in programs, put invisible commands in programs, create **Invisible File Names**, alphabetize/store info on disk, convert dec to hex or Integer to FP, protect and append programs, dump text to printer...

MORE TOO: 21 Programs Total, a best-seller!

(**$29.50**: Includes Peeks/Pokes Chart & Tip Book #3)

New!

PRONTO-DOS

HIGH-SPEED DOS / DOS-MOVE UTILITY
by TOM WEISHAAR

Put HIGH-SPEED DOS in your Apple's normal memory, Language Card or Apple IIe's high-memory—

Function	Normal	Pronto
BLOAD HI-RES IMAGE	10 sec.	3 sec.
LOAD 60-SECTOR PROGRAM	16 sec.	4 sec.
SAVE 60-SECTOR PROGRAM	24 sec.	9 sec.
BLOAD LANGUAGE CARD	13 sec.	4 sec.

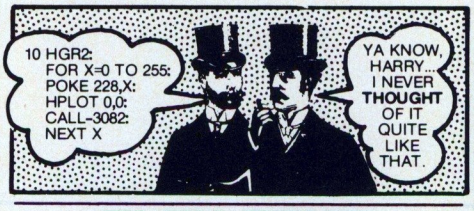
(Text Files: No Change)

Boot the Pronto disk or your updated disks, created with the normal INIT command. Compatible with ALL DOS Commands, GPLE® and most of your programs.

Move DOS to your Language Card or Apple IIe standard high-memory, freeing up **10,000** Extra Bytes!

15 Extra Sectors per disk, Catalog Free-Space display, new "TYPE" Command reveals Text File contents...

(**$29.50**: Includes Peeks & Pokes Chart)



FLEX TEXT

20/40/56/70-COLUMNS WITHOUT HARDWARE
by MARK SIMONSEN

PRINT VARIABLE-WIDTH TEXT on both hi-res screens with normal Applesoft commands (including Htab 1-70). Normal, expanded & compressed text with no hardware. 70-columns requires b/w monitor (not tv).

Add Graphics to Text or Text to Graphics. Run your existing Applesoft programs under Flex Text control. Fast, easy to use and Compatible with GPLE.®

DOS Tool Kit® font compatibility, or use the supplied Flex Text typefaces. Select up to 9 fonts with control-key commands. Custom text character editor included.

(**$29.50**: Includes Peeks & Pokes Chart)

FRAME-UP

HIGH-SPEED PRESENTATION UTILITY
BY TOM WEISHAAR

MAKE PROFESSIONAL PRESENTATIONS of existing hi-res, lo-res and text frames. *FAST* hi-res loads in 2½-seconds! Paddle or Keyboard-advance frames.

Unattended Shows are optional with each picture arranged and pre-programmed to display from 1 to 99 seconds. **Text Screen Editor** lets you create black-and-white text "slides" and add type "live" from the keyboard during shows. Mail copies of presentations on disk to friends and associates (or home to Mom!).

(**$29.50**: Includes Peeks & Pokes Chart)

New!

BEAGLE BASIC

APPLESOFT ENHANCER by MARK SIMONSEN

Requires Apple IIe (*or* II/II+ with RAM Card)

RENAME ANY APPLESOFT COMMAND or Error Message to any word, for program clarification or encryption. Plus add optional NEW COMMANDS—

ELSE follows If-Then's (as in IF X=2 THEN PRINT "Yes": *ELSE* PRINT "No"). **SWAP X,Y** exchanges 2 variables' values. Use **TONE** command to write music with no messy pokes & calls. **HSCRN** reads the color (off/on) of any hi-res dot. **SCRL** scrolls the text screen in *either* direction. **TXT2** lets Text Page 2 act like Page 1.

Also— GOTO or GOSUB may precede variables (as in "GOSUB FIX" or "GOTO 4+X". Escape-mode indicated by special ESC CURSOR. Replace Apple's awkward Graphics Screen Switch pokes with one-word commands. Change ctrl-G Beep to any tone. INVERSE REM STATEMENTS too! All GPLE® compatible.

These new functions occupy ZERO extra memory!

(**$34.95**: Includes Peeks/Pokes Chart & Tip Book #6)

BEAGLE BAG

12 APPLE GAMES by BERT KERSEY

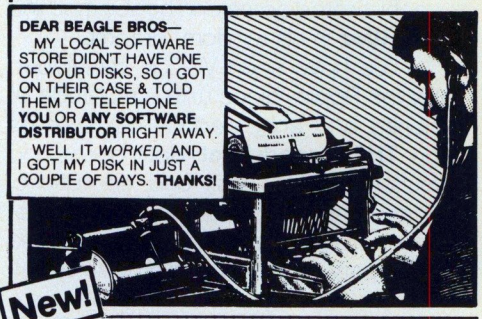
COMPARE BEAGLE BAG with any single-game locked-up disk on the market today. All 12 games are a blast, the price is right, the instructions are crystal clear, *and* the disk is COPYABLE. You can even *change* the programs or list them to learn by seeing how they work.

Twelve Games from the Applesoft Ace, Bert Kersey—TextTrain, Wowzo, Buzzword, Magic Pack & more...

Excellent Reviews— (see Jan-83 *Softalk*, page 148).

BEAGLE MENU too: Description under "Typefaces".

(**$29.50**: Includes Peeks & Pokes Chart)



New!

DOUBLE-TAKE

2-WAY-SCROLL/MULTIPLE UTILITY
by MARK SIMONSEN

LISTINGS & CATALOGS SCROLL Up *AND* Down, making file names and program lines much easier to access. Arrow-keys control scroll direction. New Applesoft LIST FORMAT: each program statement on new line for *FAST* program tracing and de-bugging.

Variable-Display shows all of a program's strings and variables with current values. **Cross-Reference** shows line nos. on which each variable & string appears. Better **Renumber/Append** to *Merge* programs (not just connect end-to-end). Also **Auto-Line-Number**, Instant Hex to Dec Converter, Program Stats, Eliminate/Redefine Cursor, Free Space-On-Disk... All GPLE® compatible.

(**$34.95**: With 2 Charts: Peeks/Pokes & "Tips & Tricks")

RUSH the following disks by First Class Mail—

☐ Alpha Plot	$39.50	☐ Flex Text	$29.50
☐ Ap.Mechanic	29.50	☐ Frame-Up	29.50
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Tip Disk #1	20.00
☐ Beagle Basic	34.95	☐ Utility City	29.50
☐ DOS Boss	24.00	☐ ADD ME to mailing list.	
☐ Double-Take	34.95	☐ ALREADY ON mailing list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA master charge

Visa/MasterCard or COD, call TOLL-FREE
Nationwide: **1-800-854-2003** ext. 827
California: **1-800-522-1500** ext. 827
Alaska/Hawaii: **1-800-854-2622** ext. 827

OR mail U.S.Check, Money-Order or Visa/MC# to BEAGLE BROS, 12th Floor
4315 SIERRA VISTA, SAN DIEGO, CA 92103

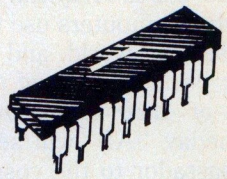
Add $1.50 First Class Shipping, Any-Size Order.
Overseas add $4.00. COD add $3.00. California add 6%.
ALL ORDERS SHIPPED IMMEDIATELY.

CLIP COUPON OR USE SEPARATE SHEET.

Circle 2 on Reader Service Card

COMPUTER FUTURES:

The Age of Silicon



Jonathan V. Post/contributing editor

A number of well-known science fiction writers have begun writing about computers. Isaac Asimov, Dean Ing, Jerry Pournelle, Vernor Vinge, James Hogan, and their compatriots have decided that truth is indeed stranger than fiction. In future articles, we hope to examine the computer/science fiction connection in greater depth, but for now: What is really going on?

Introduction

This article is the first of three, spread through three issues of *Call -A.P.P.L.E.* These articles are excerpted from a book-in-progress. You, the reader, have an input into the topics of that book, and the depth to which these topics are explained. Letters to the editor, A.P.P.L.E.-Crate, and other feedback mechanisms will let you, collectively, vote on a menu of future chapters. This menu will appear at the end of the third part of *Computer Futures.*

This first article of *Computer Futures* includes the preface (where we are now), an introductionary section, "The Age of Silicon," and a compressed outline of six topics:

- Artificial Intelligence
- Robotics
- Computer Graphics
- Personal Computers
- Exotic Technologies
- The Far-Out Future

This first article of *Computer Futures* will also include the topic of Artificial Intelligence. The second article will cover the topics of Robotics, Computer Graphics, and Personal Computers. The third article will cover Exotic Technologies, the Far-Out Future, and the menu of future options, in which you, the readers, select items for later articles. A sample of the menu-use is given, which results in a mini-essay on Distributed Computers and the Brain.

. . . ours is an interactive future. . .

Now, let's move on to the outline, and the meat of the article. As you read, note down (on paper, for you old-fashioned folks, or on your Apple) the topics, ideas, people, or references about which you would like to learn more in the future. Because — ours is an interactive future.

I Have Seen the Future . . . and It Computes!

The computer revolution is a turning point in the history of life on this planet, and it is happening right now — the most exciting time for a human being to be alive.

Before I discuss the future, and the Personal Computer Revolution of the Present, let me summarize the entire Past of human civilization in a paragraph.

We came from the stone age into the iron age. After the iron age carried us into the Industrial Revolution, we were really in the age of Carbon — oil, coal, gas & the organics. This was followed by the Age of Copper, when copper wires of telegraph and telephone linked the homes and capitals of the world. We hope to come safely through the age of Plutonium — because we want to re-enter the Stone Age.

How do we get back to the Stone Age? It's a matter of definition: stones and sand and glass are filled with Silicon Dioxide, and Silicon is the magic ingredient of the computer chip. So,

Welcome to the Age of Silicon!

The Dark Ages of Europe were replaced by the pyrotechnics of the Renaissance when painstakingly recopied manuscripts of Greek classical civilization were retrieved from monk-made "databases" and "broadcast" by printing presses. Those Greek documents amounted to little more than a billion bytes of information.

The amount of information is a hundred thousand times as large as the entire exhumed memory of Greece.

Today, the database of the laboratory, art studio, university and corporation are being broadcast by satellite into the personal computers of millions of homes. The amount of information is a hundred thousand times as large as the entire exhumed memory of Greece. Can our computerized self-discovery trigger a New Renaissance that eclipses the old one?

The computer fires self-discovery to a fever pitch. In a long-awaited decision founding a national DNA database (Science, 27 Aug 82, p. 817), the team of BBN and Los Alamos will store all known information on human genetic sequences. Perhaps someday computers will know what we're made of, and then we'll ask the computers "What is the secret of life?" And they'll say "We know what the secret of life is — but you humans aren't smart enough to understand!"

Compressed outline to fire up the reader

In this article, *you,* the reader thirsting for computer literacy will see glimpses of the future as it unfolds before your eyes . . . at a speed to keep pace with the computer itself.

—◉→

AGE OF SILICON

You'll learn how:

So-called *Artificial Intelligence* is what computers use to talk with us, see the world, and solve complex problems *beyond the capabilities of their own programmers!* Someday "AI" will serve as our ambassador to the cosmos...and as our greatest tool for self-understanding.

Robotics is more than R2D2 in Star Wars and Robbie in Forbidden Planet. Right now robots are being designed to drive our cars, cook our meals, build our televisions, and probe the alien soils of the moons of Jupiter. Someday robots (which look nothing at all like human beings) will improve our industrial productivity so much that excess leisure time will replace unemployment...and then fill our leisure time with robotic arts and sporting events.

How can we harness the energy of ten million videogame fanatics to design the new utopia?

Computer Graphics is more than just another pretty interface. One expert predicts that the computer screen *is the new home for mankind!* How can we harness the energy of ten million videogame fanatics to design the new utopia? Why is the most humongous supercomputer in California doing nothing all day but drawing pictures? What will it be like to go *through* the screen into a three-dimensional simulation *that you can touch and feel?* The secret awaits us all... behind the CRT.

Personal Computers have stuck their feet into the doors of the average home. But what will it be like when they move in to stay... *and bring their big brothers with them?* Personal computers are nothing to be afraid of, say the experts. The problem is, *they aren't personal enough.* You will discover how the personal computer is only a small step towards the *convivial computer*...which will keep us company from the cradle to the grave...*and beyond!*

Today's exotic technologies will be commonplace...

Today's *Exotic Technologies* will be tomorrow's commonplace technologies. Peek behind the locked lab doors of the highest hi-tech corporations to discover how ballistic electrons, gallium arsenide, quasi-particle quiterons, ion beams, lasers, and organic superconductors will change the very substance of computers. Find out which computers find the ice fields of Antarctica to be *ten times too hot for comfort,* and search for the even more amazing biocomputers that may ride through our veins and arteries, seeking and destroying diseased cells, *and making us immortal.*

And finally, a summary which takes you on *The Ultimate Voyage,* unleashing the restraints of your imagination as computers carry us beyond the stars...*to our true destiny.*

1. Artificial Intelligence

Not too long ago the acronym "AI" meant, to a majority of folks who used the term, no more nor less than "Artificial Insemination." After a pregnant pause, we find "AI" has come to mean "Artificial Intelligence." Those of us gifted with natural intelligence are intrigued by this new term. And yet AI is not new — the name was coined in 1957 by an interdisciplinary team of scientists meeting at Cornell.

Avron Barr and Edward Feigenbaum, at Stanford University, have defined AI as "the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior — understanding language, learning, reasoning, solving problems, and so on."

...computers can be programmed to learn, make inferences, and to create.

An exact definition of Artificial Intelligence is hard to find. AI means what AI people do when they write computer programs which they advertise as AI. Some of the things these programs do exceed the talents of their programmers. Arthur Samuel, at IBM, was the first to experience this paradox of the program surpassing its creator. He taught the computer to play checkers, in the late 1950s, and it soon was jumping over his best efforts. The old saw "computers can only do what they are programmed to do" is true as far as it goes...but computers can be programmed to learn, to make inferences, and to create.

Among the inferences and creations of the AI programs are applications including medical diagnosis (computers can match the performance of a team of trained physicians at finding tumors from x-ray images), geological prospecting (computers located a promising molybdenum deposit, and are sniffing out oil wells), chess playing (computers have won top prizes at tournaments, and provoked hostility from Grand Masters who feel threatened), and even computer design. AI is helping to design the even-more-intelligent computers of the next hardware generation.

AI is a growing field. It attracts people with diverse backgrounds — psychologists who want to test their theories on artificial personalities, linguists who want programs to actually understand language, structural engineers who want to build bridges and skyscrapers unlike any seen before, and physicists who want a new probe into the subatomic universe.

AI is on its way to being a science.

AI is on its way to being a science. As A. Newell puts it: "Scientific fields emerge as the concerns of scientists congeal around various phenomena. Sciences are not defined, they are recognized." Douglas Hofstadter's book *Godel, Escher, Bach: An Eternal Golden Braid* gives a lively account of how AI congealed from the mathematical logic of the 1930s and 1940s, the symbolic processing ideas of Church and Turing, and the Cybernetics and self-organizing systems of Wiener and McCulloch in the 1940s and 1950s.

Looming behind this history is the patriarchal figure of Charles Babbage, the man who began constructing the first programmable mechanical computer under the sponsorship of Queen Victoria, over a century ago. Why Babbage's dream was true, and why his efforts failed, provide ironic counterpoint to the tales of current success. And Charles made history in another way as well, hiring an assistant, Ada Augusta Lovelace, the daughter of Lord Byron; the woman who is regarded as the first programmer. The episode of Charles and Ada hocking the family jewels (literally) after losing a bundle on the racetrack is a cautionary tale for those who seek ill-gotten gains from their computing. Babbage mingled with the in-crowd of his day. For one of his now-forgotten plays, his lighting man was Michael Faraday. Today, there is a Babbage Museum. An even more astonishing tribute is the new Department of Defense standard computer language, named (and this is quite a surprise for Pentagon-watchers) — ADA.

How AI began, how it is proliferating, and where it may lead are the subjects of this article. Computers that talk, recognize speech, interpret visual images, and write fiction are already here. Computers that carry on a witty conversation, read between the lines, comprehend an opera, and write a best-selling novel are the goals of some researchers, and the objects of condemnations by some critics. In order to resolve these controversies, we shall be introduced to some remarkable computer programs — and their human servants.

(Among the friends I am free to interview are Professor Peter Szolovits at the AI Lab at MIT, who lived in the same high-tech commune with me back in the 60s; Oliver Selfridge, the British polymath who was once Marvin Minsky's boss, and who was the father of machine perception; Edward Riseman, chairman of Computer Science at the University of Massachusetts at Amherst, director of a computer vision project; and Rand Waltzman, another Caltech buddy, recently employed by Tecknowledge [see below] for his Expert Systems expertise)

Several large corporations have decided that AI is too valuable a field to ignore.

Several large corporations have decided that AI is too valuable a field to ignore. Among the powers-that-be who have their own AI projects are: Fairchild Instrument and Camera Corp., SRI International, TRW Defense Systems, Digital Equipment Corp., Rand Corp., Bolt Beranek and Newman, Xerox PARC, Bell Laboratories, E. G. & G. Idaho, Systems Development Corp., Amoco Production Co., Schlumberger-Doll Research, ITT Integrated Systems Center, and the Electrotechnical Laboratory Japan. AI is routinely projected to be a billion dollar business.

Also of great interest to the financial community are three new private companies, each spun off from universities where the outcast hacker of yesterday is the high-caste consultant of today. Cognitive Systems, Inc., spun-off by Roger Schank, director of the Artificial Intelligence Laboratory at Yale, is based on the first language-independent meaning representation embedded in a program. Schank's system stores complex bundles of meaning in a way which does not depend on English, and could (time will tell) just as easily handle Russian, Japanese, or extra-terrestrialese. Language translation by computer is, in fact, a way of life at places such as Brookhaven National Laboratory, where a quick and cheap English version of a Russian physics paper is preferred to the slow, expensive, and subjectively distorted versions done by human beings.

Language translation by computer is, in fact, a way of life...

IntelliGenetics, spun off from Stanford in 1980, simulates almost any possible gene-splitting experiment, giving researchers the nucleotide sequences and restriction maps that make genetic engineering possible. The Science Fiction dream of manipulating the human chromosome will depend on such computer systems — no human being could handle the immense database of the genome with a seat-of-the-pants approach. IntelliGenetics founders include Douglas Brutlag, reknowned expert in DNA structure and synthesis and enzymology; Edward Feigenbaum, who is believed by many to be the founder of the field of Knowledge Engineering; and Laurence Kedes, an animal-cell virtuoso. IntelliGenetics advises on large scale industrial fermentation projects, but what they are brewing in their vats can not be found in your neighborhood bar.

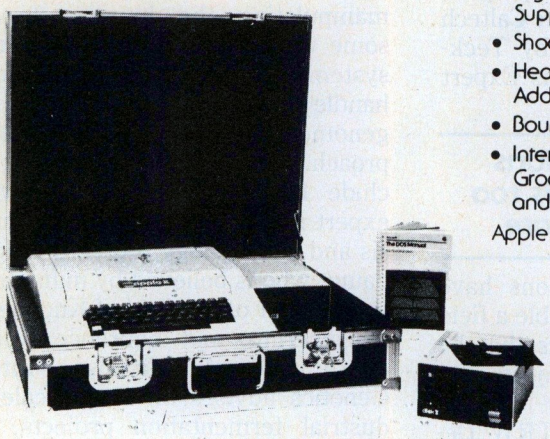
Tecknowledge, Inc., started in 1981 by a team from MIT, the Rand Corp., and Stanford, claims that its staff of 40 comprises a third of the world's top experts in the design and operation of knowledge systems. They are in the business of leading Ph.D.s by the hand through the knowledge engineering biz while corporations pick up the rather hefty tab.

The first of the knowledge engineering systems, a.k.a. Expert Systems, was Dendral, born 1965 to Joshua Lederberg (Nobel Laureate from Stuyvesant High School, my old stomping ground) at Stanford. Dendral infers the structure of organic molecules from mass spectrogram data, and is used by chemical companies world-wide. Dendral once was matched against a super-chemist known as "The Mozart of Mass-Spectrometry" — and won.

—◉→

FLIGHT FORM

BY CASES, INC.
DESIGNED TO PROTECT YOUR COMPUTER



Features —
- Rigid Shell Made of Plywood Supported High Impact ABS.
- Shock Resistant Foam Lining.
- Heavy Duty Hardware Add $22.50 (U.S. dollars) for lock.
- Bound Metal Edges.
- Interlocking Tongue and Groove Extrusion, Mating Lid and Bottom.

Apple II Plus 2 Drives $250.00

OTHER MODELS AVAILABLE

Overseas Inquiries Welcome

CLUB AND DEALER DISCOUNTS AVAILABLE

CASES, INC. P.O. Box 33820 Seattle, WA 98133-0820 (206) 365-5210

VISA
MASTERCARD

Circle 33 on Reader Service Card

SILICON

Knowledge Engineering, which is the buzzword for AI backed by big bucks, can be used for electronic equipment failure diagnosis, computer systems configuration (The R1 program at DEC designs for the customer a unique layout of VAX computer options, down to the cables and screws), VLSI design (the Very Large Scale Integrated circuits of the modern computer) have the intricacy of a San Francisco street map, squeezed onto a matchbook cover, Ultra Large Scale Integrated circuits (ULSI) will be as complex as a complete United States of America street map. By the late 1990's, exotic technology devices will store the equivalent of a map of the Earth to one centimeter resolution (ten to the eighteen bit, a "quintilla-bit." People can't do the designing without a helping hand), crisis management, and military threat assessment and targeting. For more on the military implications, see my May 1979 OMNI article: Cybernetic War.

This ends our discussion of the topic "Artificial Intelligence." Next issue of *Call -A.P.P.L.E.* will begin with a brief summary of what has been covered so far, and then launches into an investigation of Robotics, Computer Graphics, and Personal Computers.

Don't erase your notes on items you'd like to learn more about.

We'll interact again, soon.

"BACK-UP" YOUR DISKS
ESSENTIAL DATA DUPLICATOR

The most powerful disk duplicator available for your Apple® computer.
EDD can "Back-Up" much more protected Apple software than all of your other copy programs put together.
Since EDD is almost totally automatic, no longer will you have to put up with changing parameters to duplicate most disks, although every parameter is fully documented in our extensive manual. We also are providing updated program parameter lists.

$79.95
CA residents add 6%

Runs on: Apple II, II plus IIe, or III (emulation mode with 1 or 1, 3.3 drives

- EDD rarely needs parameter changing
- Automatically finds the beginning of each track
- Accurately finds "Auto-Sync" bytes and their lengths
- Can copy ¼ tracks
- Plus many more features

TO ORDER OR FOR MORE INFO, CALL 707-257-2420
UTILICO SOFTWARE
3377 Solano Ave., Suite 352, Napa, CA 94558

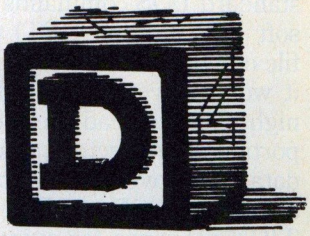
US

Circle 25 on Reader Service Card

```
]LIST

5  REM

     SO WHAT DID YOU
       EXPECT

GARTH BAULCH : CALL
     -A.P.P.L.E.

10   HGR : TEXT : HOME
       :
20   HTAB 9: VTAB 12:
       PRINT "SO WHAT
       DID YOU EXPECT?"
30   FOR A = 1 TO 24
40   READ X
50   POKE 767 + A,X
60   NEXT A
70   CALL 768
80   DATA  44,87,192,44,82,19
       2,169,96,141,80,192,23
       4,234,76,16,3,141,81,1
       92,234,234,76,8,3
```

D IS FOR DATA:

Where Has All The Data Gone?



Ken Kashmarek/Call —A.P.P.L.E. staff writer

Introduction

AFTER diligently entering the contents of my checkbook into Personal Finance Manager (PFM), a package which came with my Apple][Family System, I attempted to process the data with an Applesoft program. Much to my dismay, END OF DATA (horrors) was encountered on the very first INPUT. Where has all the data gone (long time passing . . .)?? It was almost the end of the world, or at least, the beginning of another.

More appropriately, the above question should be stated: "How is the data stored?." This plagued me for some time after the END OF DATA experience. The data was intact and PFM had no problem using it. According to the documentation, the data was stored on a standard DOS text file. I soon determined that Applesoft Basic programs could not process this file. Why? Over the next year, that question was pursued.

During that pursuit, I not only answered the question, but put together some programs and techniques for disk based file handling using Applesoft and DOS 3.3 on the Apple][. This series, "D Is For Data," will cover data storage and retrieval techniques for all file types. This issue will cover my nemesis, PFM.DATAFILE, and the data formats used by Personal Finance Manager. Followup articles will cover file access methods using the DOS file manager (via ampersand routines), data handling techniques, and eventually a utility for processing the data in PFM.DATAFILE. Stay tuned. You are about to move into the world of data processing.

The specific objective of "D Is For Data" is to introduce data storage and retrieval techniques to the Apple][environment. PFM.DATAFILE will be used as an example of how to retrieve, process, and store data in the internal binary formats that are used by Applesoft. These techniques provide for efficient storage and processing of data.

Personal Finance Manager

Personal Finance Manager is a friendly, easy to use checkbook balancing program. It will also keep track of cash entries, up to 12 credit card accounts, and up to 24 budget accounts. PFM provides for data entry, update, retrieval, reporting, printed reports, and Low-Res graphics displays. It will handle up to 200 checks (or entries) per month, can reconcile your checkbook, and roll the file forward at the end of a year (allows for up to 200 unreconciled entries to be carried forward to the new year). All together, PFM is a well done product.

With such a glowing buildup, what is the problem? To quote David A. Lingwood, *"Thou shalt not diddle with the data"!* So what? PFM.DATAFILE is a DOS 3.3 text file. What's the big deal? That is what I asked Apple Computer when I reported a problem with being unable to process the file with Applesoft. The first answer was a courteous "Thank you for your question". Period. There was no reply when the question was asked a second time. Lack of a response indicated that all was not well with the data. Or at least, the data was not in the format that was indicated by a text file.

The glaring omission is lack of documentation on the data file, and how it is used. In fact, PFM documentation states that PFM.DATAFILE is a normal DOS file with all the properties thereof (page 56 of the manual). Since it is allocated as a text file, one might assume the properties of a text file. This turns out to be an invalid assumption.

PFM.DATAFILE Characteristics

Having discovered that I was on my own, *Bag Of Tricks* (Quality Software) was used to dump the file, and determine why it could not be read with Applesoft. The accepted format of a DOS text file is that each data byte represents an ASCII character stored with the high bit on, data records terminated by a carriage return ($8D), and the file ends with a zero byte ($00). A random text file allows the zero byte to end each record also.

PFM.DATAFILE has none of these attributes. In fact, PFM.DATAFILE contains binary data entirely. Character fields do not have the high bit on, nor are they terminated by carriage returns. Floating point values are stored in Applesoft internal format (five bytes: one byte exponent and 4 byte mantissa). Integer values are stored in the internal integer format (HI/LO byte pair). All unused bytes are set to zero. In my case, over 60% of the file is zeros. Now, that is some text file! Perhaps we should ask, where has the standard gone? Why would a text file (DOS file type = 0) be used to store binary data?

PFM.DATAFILE is a very long binary file, without the standard binary file pointers in the first four bytes. This brings up the 64K question: How can this file be processed from Applesoft? Not very easily as it turns out, but it can be done. Processing PFM.DATAFILE from Applesoft can even be done without assembly language routines, although it would take a pile of PEEKs and POKEs (or, as they say back on the farm, a whole spreader load).

WHERE HAS ALL THE DATA GONE?

My first attempt was to use the standard DOS commands and Applesoft INPUT (and GET) to read the file one byte at a time. I estimate that it would have taken 40 days and 40 nights just to read the file. The data portion is 278 sectors, or over 70K of data. With over 60% zeros, and with a DOS CLOSE, OPEN, and POSITION sequence for every zero byte, you can appreciate the problem. The next attempt was to read the file a sector at a time using "L256" on the OPEN command. This worked to get that data into memory, but needed that manure spreader full of PEEKs and POKEs to move the data from the DOS data sector buffer to where Applesoft could use it. In order to answer the question about storing binary data in a text file, it was time for some research.

DOS 3.3 File Processing

DOS 3.3 supports several different file types. Only the text file is available to be processed directly by BASIC programs written in Applesoft. Text files are accessed with OPEN, CLOSE, READ, WRITE, POSITION, and APPEND commands via the DOS intercept ([Ctrl-D] recognized at the beginning of a line of output). The data is read into memory using INPUT (or GET) and written to disk using PRINT. This technique is essentially "pseudo" keyboard input and screen output redirected from and to the disk via the DOS intercepts. As indicated earlier, text files have special attributes, none of which are used by PFM. DATAFILE.

Applesoft programs can also process binary files via the DOS intercepted commands [Ctrl-D] BLOAD and [Ctrl-D] BSAVE. However, the entire file must fit in memory, and the length must be less than 32K bytes. Processing binary files in this fashion is essentially a memory to memory technique.

This is the extent of file processing options directly available to an Applesoft program. Other file types are restricted to program storage (Integer and Applesoft), or are undocumented ("S", special "A", and special "B"). The "R" type file is in use by the DOS Tool Kit Assembler (Apple Computer) for relocatable binary files (assembly language programs). At this point, a text file was chosen probably due to the need to store over 70K bytes of data. While Applesoft can process text files, there is some difficulty in processing the binary data.

This does not mean that PFM. DATAFILE is not accessible through Applesoft. It just means that we cannot use the standard available techniques. A point should be made here. The Applesoft Basic language has no file handling capabilities itself. Data on disk files are made available to Applesoft and Integer programs via DOS intercepted commands. The intercept technique does not allow for efficient processing of data files by Applesoft programs. In fact, Applesoft is unaware that the data is coming from or going to disk. With this in mind, we must take a closer look at PFM.DATAFILE in order to determine how to process the data.

The PFM.DATAFILE Format

Figure 1 presents the general layout of the file. The total length is 281 sectors, of which 278 sectors are for data. Three sectors are track/sector list pointers which point to the data sectors (each track/sector list contains 122 pointers). The data portion is 70K bytes, which is too large for a binary file, and too large to fit in memory. This is probably the reason for choosing the text file for storing the data, since a text file is limited only by the size of a diskette.

PFM.DATAFILE is broken down into three sections. The base data area is 18 sectors long (4608 bytes), and contains all of the primary file information. This includes monthly totals (number of checks and dollars spent), budget definitions, budget expenditure summaries, and credit card expenditures. When PFM is booted, this portion of the file is loaded into memory until the QUIT option (10 on the menu) is chosen to end the PFM program. At that point, it is written back to disk.

The next section of the file is 20 sectors long, containing the year end carryover entries (unreconciled checks). The format of this section is identical to the monthly data sections.

The final section of the file is 12 groups of 20 sectors each, one group for each month beginning with January, and ending with December. PFM is supposed to handle 200 entries per month, but in fact only stores 199 entries per month (entry number 0 for each month is not used). Each sector stores 10 entries of 25 bytes apiece, with six unused bytes at the end of the sector.

Each sector of this file is reserved for specific data items. Each byte or data field in each sector has the function of storing a specific data value. In PFM.DATAFILE, a field is considered a data item. Each field is of fixed length and at a fixed position in the file. With this technique, a specific data item, current balance for example, is always at the same location in the file. The file does not have to be processed from the beginning in order to find the current balance. The sector is read into memory (in this case, the entire 18 sector base data area), and the location of the current balance is calculated (beginning memory address of sector #0 plus 936).

The monthly data sectors contain reserved fields for 10 data entries per sector. Once a data sector is read into memory, each data field can be accessed directly. This technique provides for rapid access of the data. Field separator characters are not necessary, since the length and location of each field is known. The size of the file is always the same, and space for future entries is reserved within the file.

All data is stored in binary format as used in memory by Applesoft. Floating point numbers take five bytes, integers take two bytes, while character values are generally 20 bytes apiece (file title is 22 bytes). Characters are stored with the high bit of each byte off (no carriage return characters).

Figure 2 presents the detailed layout for the base data area. More than half of this area is for budget definitions and expenditures. This data is processed by reading 18 consecutive sectors into 18 consecutive pages of memory. Some of the fields are marked "use unknown." These fields may have some significance within PFM itself, but I have not been able to determine that significance. Figure 2–A is a hex and character dump of sector #0.

Figure 3 presents the layout of each individual entry (a check, for example) in the monthly data sectors. Each data sector contains 10 entries of 25 bytes apiece (six unused bytes at the end of a data sector). Empty data sectors have $8D in the first byte followed by 255 bytes of zeros ($00). Figure 3–A is a hex and character dump of sector #58 (February entries). Note that several data items are packed into single fields.

Check numbers range from 1 to 9999. The check number field is also used as an indicator for credit card entries (–1 to –12), cash expenditures (–0), or deposits (–13). The high order bit of the day and month fields store information about reconciled checks and tax deductable entries, respectively. Budget id., check notation, and dollar amount take up the major portion of the entry.

The monthly data entry sectors are processed one sector at a time. During data entry, the sectors are written out as they are filled. When entries are updated, the sectors are read and written as each entry is changed. During a search process, the data sectors are scanned from beginning to end, unless restricted by search criteria (specific months or check numbers).

At this point, keeping data in a binary format on a disk file may be difficult to comprehend. Let's look at it from two points of view. First, the space considerations. The capacity of this file is 2587 entries (199 entries per month times 13 months). Each entry takes 25 bytes for a total of 64675 bytes. The same information stored in character format would take approximately 42 bytes per entry, for a total of 108654 bytes (68% larger). A normal text file processed by INPUT must have each data item separated by carriage returns, or separated by commas with quotes around character strings (using a single INPUT command to read in the entire entry). In the standard text file definition, these extra characters generate much of the additional space requirement. Check number would take five bytes and the dollar amount would take six, seven, or eight bytes to store in a standard text file.

Secondly, text file processing via the DOS intercept is inefficient. When a file is processed, the data is handled by DOS one character at a time (through INPUT or PRINT). An Applesoft program is slowed down so much that disk revolutions may be missed when attempting to access subsequent sectors of the file. In addition, the ASCII character data is converted to Applesoft internal form on INPUT and vice versa for PRINT. This processing further slows down the Applesoft program. While this slowdown may not be noticeable for small files, it is immediately obvious on large files.

So far, we know why a text file is used, what that data looks like, and how it is stored on disk. Now, how can that data be processed?

Tune in again next month, same time, same place...

This issue of "D Is For Data" has taken you through Personal Finance Manager, a touch of DOS, and a detailed look at the format of PFM.

```
0000:00 00 00 45 00 42 00 4F  ...E.B.O
0008:00 53 00 40 00 45 00 46  .S..E.F
0010:00 50 00 43 00 46 00 3D  .P.C.F..
0018:00 3E 0E 8B 00 01 00 C8  ........
0020:00 01 00 0C 00 1F 00 01  ........
0028:00 0C 00 00 00 00 FF FF  ........
0030:4B 41 53 48 4D 41 52 45  KASHMARE
0038:4B 20 31 39 38 32 20 20  K 1982
0040:20 20 20 20 20 20 00 00          ..
0048:43 43 43 52 45 44 49 54  CCCREDIT
0050:20 43 41 52 44 20 50 41   CARD PA
0058:59 4D 45 4E 54 53 00 00  YMENTS..
0060:46 45 46 45 44 45 52 41  FEFEDERA
0068:4C 20 49 4E 43 4F 4D 45  L INCOME
0070:20 54 41 58 20 20 00 00   TAX  ..
0078:49 41 53 54 41 54 45 20  IASTATE
0080:49 4E 43 4F 4D 45 20 54  INCOME T
0088:41 58 20 20 20 20 00 00  AX    ..
0090:46 49 53 4F 43 49 41 4C  FISOCIAL
0098:20 53 45 43 55 52 49 54   SECURIT
00A0:59 20 20 20 20 20 00 00  Y     ..
00A8:48 50 48 4F 55 53 45 20  HPHOUSE
00B0:50 41 59 4D 45 4E 54 20  PAYMENT
00B8:20 20 20 20 20 20 00 00          ..
00C0:43 50 43 41 52 20 50 41  CPCAR PA
00C8:59 4D 45 4E 54 20 20 20  YMENT
00D0:20 20 20 20 20 20 00 00          ..
00D8:45 4C 50 41 52 4B 20 56  ELPARK V
00E0:49 45 57 20 45 4C 45 43  IEW ELEC
00E8:54 52 49 43 20 20 00 00  TRIC  ..
00F0:57 41 50 41 52 4B 20 56  WAPARK V
00F8:49 45 57 20 57 41 54 45  IEW WATE
```

Figure 2A - Sector #0

```
0000:8D 00 00 00 00 00 00 00  ........
0008:00 00 00 00 00 00 00 00  ........
0010:00 00 00 00 00 00 00 00  ........
0018:00 85 50 A3 D7 0A 0C 51  ..P....Q
0020:02 81 54 56 43 4F 58 20  ..TVCOX
0028:43 41 42 4C 45 20 20 20  CABLE
0030:20 20 87 22 1E B8 52 0C    ....R.
0038:52 02 81 20 20 47 45 20  R..  GE
0040:52 45 50 41 49 52 20 20  REPAIR
0048:20 20 20 86 21 0A 3D 71      .....
0050:0C 53 02 82 47 52 53 55  .S..GRSU
0058:50 45 52 20 56 41 4C 55  PER VALU
0060:45 20 20 20 83 31 47 AE  E   .1G.
0068:15 0C 54 02 85 47 52 53  ..T..GRS
0070:55 50 45 52 20 56 41 4C  UPER VAL
0078:55 45 20 20 20 84 36 3D  UE   .6.
0080:70 A4 0C 55 02 86 20 20  ...U..
0088:56 45 4E 54 55 52 45 20  VENTURE
0090:20 20 20 20 20 20 86 4D        .M
0098:F5 C2 90 0C 56 02 86 20  ....V..
00A0:20 42 52 41 55 4E 53 20   BRAUNS
00A8:20 20 20 20 20 20 20 85         .
00B0:2D 0A 3D 71 0C 57 02 86  .....W..
00B8:20 20 50 45 54 45 52 53    PETERS
00C0:4F 4E 53 20 20 20 20 20  ONS
00C8:84 3D 99 99 9A 0C 58 02  ......X.
00D0:86 20 20 48 41 49 52 20  .  HAIR
00D8:50 45 52 46 4F 52 4D 20  PERFORM
00E0:20 85 24 B8 51 EB 0C 59   ...Q..Y
00E8:02 86 20 20 4C 45 52 4E  ..  LERN
00F0:45 52 20 20 20 20 20 20  ER
00F8:20 20 00 00 00 00 00 00    ......
```

Figure 3A - Sector #58

WHERE HAS ALL THE DATA GONE?

DATAFILE. In particular, we have learned that not all files are what they seem, and that there is method to the madness of storing binary data in a text file. You will find that many data management and data base systems store binary data rather than text data (DOS definition). This technique is a necessity for efficient storage and processing of large amounts of data.

What we have not done is access the file and process the data. Next month, "D Is For Data" will provide ampersand file access routines for Applesoft Basic programs to read and write PFM.DATAFILE, plus all other file types supported by DOS (and I mean *all*). The emphasis will be on accessing disk data files. Look out FID, here we come!

PFM.DATAFILE Layout

Application:		Personal Finance Manager
File name:		PFM.DATAFILE
Allocation:		DOS 3.3 TEXT file (type = 0)
Total file size:		281 sectors
Track/sector lists		three sectors
Data		278 sectors
Sector #	Size	Description
0– 17	18	Base data area, system parameters, budget and credit card information, summary data
18– 37	20	Carryover transactions from previous years data (unreconciled checks)
38– 57	20	January entries
58– 77	20	February
78– 97	20	March
98–117	20	April
118–137	20	May
138–157	20	June
158–177	20	July
178–197	20	August
198–217	20	September
218–237	20	October
238–257	20	November
258–277	20	December

The data is addressed by relative sector number in the disk file. The base data area is an image of the data as it is kept in memory while the file is being processed. This area in memory is 4608 bytes long. The remaining data sectors are processed one sector at a time.

Figure 1.

Base Data Area Layout (as held in memory)

Byte offset		Description
0–	1	Number of previous year entries
2–	25	Number of entries per month for January through December. Each entry is 2 byte integer, stored in Applesoft integer format (HI/LO)
26–	27	Next check number
28–	29	Printer slot
30–	31	The value 200, maximum number of checks per month
32–	33	$0001 (use unknown)
34–	35	Current month number (1–12)
36–	37	Current day of month (1–31)
38–	39	$0001 (use unknown)
40–	41	$000C (use unknown)
42–	43	$0000 (use unknown)
44–	45	$0000 (use unknown)
46–	47	$FFFF (use unknown)
		All of the above fields are stored as Applesoft integer values, in HI/LO form (most significant byte first)
48–	69	User specified file title (22 bytes)
70–	71	$0000 (use unknown)
72–	647	24 entries of 24 bytes each (576 bytes). Each entry represents a budget category: 2 bytes budget ID 20 bytes budget name 2 bytes ($0000)
648–	935	12 entries of 24 bytes each (288 bytes). Each entry represents a credit card category. The layout is identical to the budget layout.
936–	940	Current checkbook balance stored in Applesoft floating point form (five bytes).
941–1000		12 entries of five bytes each (60 bytes). This is the dollars spent per per month.
1001–2440		12 months of 24 entries at five bytes each (1440 bytes). Dollars spent per month for each of the 24 budget categories.
2441–3880		12 months of 24 entries at five bytes each (1440 bytes). Dollars budgeted per month for each of the 24 budget categories.
3881–4600		12 months of 12 entries at five bytes each (720 bytes). Dollars spent per month for each of the 12 credit card categories.
4601–4607		Unused.

All character fields are stored with the high order bit off.

Figure 2.

Monthly Check Entries

The data portion of the file will hold 200 entries per month, including the previous year's entries. Each month is stored on 20 sectors of disk, with 10 checks or entries per sector. Each entry is 25 bytes in length, with six unused bytes at the end of each sector. Empty sectors (no data) contain $8D in the first byte followed by 255 bytes of $00. The format of a check entry is:

0– 4	dollar amount (Applesoft floating point)
5– 6	check number (integer)
7	month (value 1–12) with high order bit on if the dollar amount is tax deductable.
8	date (value 1–31) with high order bit on if check is reconciled.
9–10	budget id for this entry
11–24	check notation

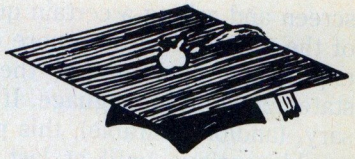
The maximum check number is 9999. If the check number is negative, then the absolute value is an indicator, not a check number:

–1 to –12	credit card entry 1 to 12
–13	deposit
–0	cash entry

Figure 3.

AN APPLE FOR THE TEACHER:

BASIC vs LOGO



Lee Reynolds/Call —A.P.P.L.E. staff writer

RECENTLY I acquired Terrapin Logo, and for about the past two months, I have been playing around with the language. I am definitely not as proficient in it as I am, for example, in Applesoft (after all, I've programmed in Applesoft for over three years), but at least I have enough familiarity with the language to write an article about it. I decided to compare Logo with BASIC (mostly Applesoft) because almost every Apple owner in the club will probably have some proficiency in that language, and I think a good way to introduce something new is to contrast it with something old...

This article will only deal with Terrapin Logo, but most of its comments and facts also apply to the other varieties of Logo available for your favorite computer: Apple Logo, and Krell Logo. If you want to read more detailed information regarding how the other varieties of Logo compare to Terrapin's, you should run, not walk, to your local library, or otherwise obtain a copy of the August, 1982, issue of *Byte* magazine. This particular issue is almost entirely devoted to Logo. (Not only for the Apple, but also for the TI-99 computer and the Trash-80).

Logo is one of the newest crazes in the programming world. Like Pascal, it is fast developing a vocal group of fans who think it's the greatest thing to come along. The adherents of Logo especially predominate among educators, because the language has been planned to be both relatively simple to learn and yet potentially quite powerful. Like Pascal, it is supposed to teach you good programming habits as you learn it. For example, it is a "structured" language. More about that later on.

Unlike Pascal, it is not a compiled language — it's interpreted like BASIC. (Personally, I have a vast prejudice against compiled languages, because there are so many tedious steps you have to go through in order to a get a working program. You create it, compile it, link it, execute it, find it doesn't work, so you re-create it with some "debugging" type of statements in it — print the value of this variable and that variable so you can understand where the program is going wrong. Then compile it again, link it again, execute it again, find you still don't have enough information to figure out where the program is messing up, so you edit the program again, with more debugging type of statements in it, compile and link once more... Ad nauseum...On the other hand, with an interpreted language like BASIC, you just insert easily-removed STOP statements at the proper places, and print out what variables you need to determine what the program is actually doing rather than what you *thought* it should be doing when you first wrote your masterpiece of logical efficiency...No compiling, linking, just edit and RUN...)

Also, the error messages are much easier to understand than Pascal's or BASIC's. Are you disenchanted with the ambiguousness of that bane of all BASIC programmers: "(Beep!) SYNTAX ERROR"? "*Now* what did I do wrong," you ask yourself. Then you will probably be very appreciative of Logo's generally more detailed and definite explanations of how you goofed in your programming...

Well, let us commence with a more or less detailed comparison of Logo and BASIC. First of all, some general comments...

Logo is a Structured Language

"Structured Programming" is all the rage these days. Personally, I think some of the principles of Structured Programming are carried to extremes at times ("Ban the GOTO statement!" "All subroutines should have only one entrance and only one exit!"), but I can appreciate a great many of the arguments in favor of it. For example, "top-down" design is usually a very logical way of approaching the task of developing a program. It also makes checkout much easier, generally.

"Top-down" design means you write a program in terms of iteratively smaller and more detailed steps. What must my program do? On the most abstract level, it inputs certain values, then computes certain other values on the basis of those input, then outputs the desired functions of the input values. So, you might decide to have a routine called INPUT which obtains the basic values needed, then a routine called COMPUTE which performs the required computations, then a routine called OUTPUT which displays the results. The main program, if expressed in terms similar to Integer Basic where you can have named subroutines, would have only the following statements in it:

```
GOSUB INPUT
GOSUB COMPUTE
GOSUB OUTPUT
```

(Of course, not all these statements would be legal even in Integer Basic, since the names of subroutines cannot be or contain any reserved words of the language, like INPUT.)

BASIC VS LOGO

Then each of the subroutines would divide the tasks of inputting, computing, or outputting, into more fundamental tasks (like clearing the screen and asking a certain question of the user), which are more closely expressible in terms of the basic statements of the language. If necessary, (and/or desirable), this process would continue until at last everything could be coded directly in the language.

Another byword of "Structured Programming" is "No GOTO's!" The constant refrain is "eliminate spaghetti programming," programs which are a maze of GOTO statements that are nearly impossible to understand. Certainly it is true that the minimization — if not the total elimination — of "GOTO" type statements is often commendable in terms of comprehensibility. My impression is that a language like Pascal is *too* strongly structured. An overly strict adherence to the principles of structured programming can be very restrictive and tedious...(That is my opinion only. I'm sure some people will disagree with me, and not without validity.)

How is Logo "structured"? For one thing, there are no line numbers. If a particular statement in a Logo program must branch to another statement (yes, there is a GOTO-type statement in the language), then you assign a label to the statement branched to, never mind the other statements of the program. (The naming conventions for statement labels are the same as those for ordinary variables. Names do not have to begin with letters of the alphabet, they may be any length and may contain just about any typeable character.)

Another way in which Logo is structured is that it is what is called a "procedural" language. That means the programs you write are made up of "procedures" (which roughly are equal to subroutines.) Even the entire program itself is a procedure, which calls its subprocedures, if any. Procedures are of two main types: what a FORTRAN programmer would call "functions" and "subroutines". A function-type procedure returns a single value based on its input(s); it does this by means of the OUTPUT statement, which also results in an immediate "return" to the calling procedure. A subroutine-like procedure, on the other hand, does not return a value, and relinquishes control to the calling procedure by means of the STOP statement.

Still another example of Logo being structured is the availability of an IF...THEN...ELSE statement, so that you can specify in one statement both what actions are to be taken when a condition is true and when it is false. (This reduces the need for GOTO's, which horrify any true aficianado of structured programming beyond words...)

A final example of Logo's structured programming concepts is that of "global" versus "local" variables. All inputs to a procedure are local to that procedure. For example, consider this procedure:

```
TO SUM :X :Y
OUTPUT :X + :Y
END
```

The name of the procedure is SUM. (You define the procedure by using the keyword "TO", followed by the procedure name, then any inputs it requires.) The two inputs in this case — what other languages often call "arguments" — are X and Y. These two names could be used as inputs to other procedures and apply to entirely different values or things. In addition, if procedure SUM were modified so as to call another procedure using, say, X as one of the inputs, that second procedure could modify the value of X, but the calling procedure, SUM, would never know it! That is because each procedure has its own private library of names, and the main way in which procedures are to communicate with another is through inputs and outputs.

However, any name used in a procedure that is not an input may belong to the private library of the procedure which called it, or to any other along the chain of calls back to the main procedure. Names which are first used in the main procedure are global names, and thus available to all its called procedures and so on down the line.

Logo's Arithmetical Capabilities

Logo, like Applesoft, allows both integer numbers and floating point numbers, but there is no naming convention which differentiates them. In other words, an integer name doesn't have to be followed by a percent sign, unless you personally want to do it that way. Also, integers are expressed in four bytes, not two. That means the maximum allowable integer is not our old friend 32767, but rather 2 to the 31st power minus 1, or over 2 billion!

Most of Applesoft's useful arithmetic functions are available in Logo. All of the trigonometry functions, like SIN and COS, use arguments expressed in degrees, not radians. (I find that much more convenient. Who but a math major ever remembers that 90 degrees is equal to pi/2 radians?)

There are no SGN or ABS functions, but they are easily enough defined as procedures. For example, you could define the absolute value function like this:

```
TO ABS :X
IF :X < 0 THEN OUTPUT -1 * :X
ELSE OUTPUT :X
END
```

You might be wondering what those colons in front of X are doing in the procedure. These signify that the procedure uses the VALUE of X, not the name X itself. Thus, a statement which some beginning programmers find confusing in BASIC, namely:

```
X = X + 1
```

would be translated into Logo like this:

```
MAKE "X :X+1
```

I'm not sure why people would find the BASIC statement hard to understand, unless they confuse the "equals" sign with the same sign used in Algebra; it is certainly true that there is no (non-infinite) number which would satisfy the "equation" X=X+1. In the Logo equivalent, the keyword MAKE is somewhat like the keyword LET which you can use in BASIC: LET X = X + 1. You distinguish the NAME X (not the value of X) by preceding it with a single quotation mark.

Logo has a function called INTEGER which corresponds to Applesoft's INT, and a function RANDOM corresponding to RND. (There is also a RANDOMIZE statement.) There is a function REMAINDER which is like Integer Basic's MOD function, and a function QUOTIENT which returns an integer division value (that is, no decimal fractions.)

However, one big fault in Logo, in my book, is the absence of a raise-to-a-power capability. Similarly, there are no functions corresponding to Applesoft's LOG or EXP.

Conditional Tests

You've already read about the IF...THEN...ELSE statement. There's another way you can do that, which is sometimes more convenient. Instead of the IF statement in the example procedure ABS, you might write it this way:

```
TEST :X < 0
IF TRUE THEN OUTPUT -1 * :X
IF FALSE THEN OUTPUT :X
```

The keywords "IF TRUE" and "IF FALSE" can also be abbreviated as "IFT" and "IFF," respectively.

Logo has the capability of defining things called Predicates, which are procedures that return a value of True or False based upon the input. The built-in Predicates all end with a question mark as part of the name. One of these is NUMERIC? which will tell you whether its input is a number or not. Another is THING? To understand the strange name of this predicate, you have to realize that Logo people do not often speak of the "value" of a "variable" but rather the "thing" of a "name." Thus THING? will return a value True if its input has a value associated with it.

When you want to test more than one condition, you don't use AND or OR, you use built-in logical functions called ALLOF and ANYOF. Here are a couple of Logo examples with their BASIC equivalents:

1. Logo:
   ```
   IF ALLOF (:X < 0) (:Y >= 0)
   THEN MAKE "Z :X + :Y
   ```
   Basic:
   ```
   IF X < 0 AND Y >= 0 THEN Z
   = X + Y
   ```

2. Substitute "ANYOF" in the Logo example above, and "OR" in the Basic example.

Loops

What non-trivial program doesn't have a loop in it somewhere? It's almost as if one of the prerequisites for making a computer program is that it involves something being done over and over again, many times. In BASIC, you have the FOR...NEXT construction, or the GOTO-type of loop, where you jump back to a statement executed previously (usually as a result of an IF statement being satisfied.) In Logo, you can also do the latter, but the keyword is "GO", not "GOTO". The former type of loop is a bit different. I will give you an example using the function "factorial n." (How many of you remember that from your high school algebra? Factorial N, where N must be a positive number, is defined as the product of N itself and all lesser positive numbers; the only exception is factorial zero which is defined to be 1. Thus factorial 3 is 3 times 2 times 1, or 6.)

```
10  REM BASIC EXAMPLE TO
    COMPUTE FACTORIAL N
20  NFACT = 1
30  FOR I = 1 TO N
40  NFACT = NFACT * I
50  NEXT I
```

The Logo equivalent might be this:

```
MAKE "NFACT 1
MAKE "I 1
REPEAT :N
  [MAKE "I :I + 1
    MAKE "NFACT :NFACT * :I]
```

The REPEAT statement will repeat whatever statements are contained between the left and right square brackets the specified number of times. This is not quite as powerful as BASIC's FOR loop, because you cannot modify a loop variable ("I," in the example) in order to exit the loop early. Also, there is no capability for STEPping with an increment other than 1.

Recursive Procedures

When you start programming in Logo, you will have to modify your way of thinking about how to solve problems. This is true in more ways than one. One of these ways is that you start thinking "recursively" instead of "iteratively." ("Iteratively" means doing something by iterating, or repeating, over and over again.) This is because Logo's REPEAT statement is not as powerful as BASIC's FOR loop, but also because Logo allows recursive procedures. What are recursive procedures? These are procedures that "call themselves." Here is a recursive solution to the compute-factorial-n example above:

```
TO FACTORIAL :N
IF :N > 0 THEN OUTPUT :N *
FACTORIAL :N -1 ELSE OUTPUT 1
END
```

Let's take a look at how this much shorter and more elegant solution works: First, note that if N is zero, then the procedure will immediately return with a value of 1 (zero factorial is defined equal to one, remember?) Suppose N is 3. Then the IF statement will be satisfied, so the value returned should be the product of 3 and factorial 2. The value of factorial 2 is not known on the first entrance to this procedure, so it will call itself in order to compute the value of factorial 2. This time the IF statement will want to return a product of 2 and factorial 1. But what is factorial 1? The procedure calls itself again in order to find out. Now we will return the product of 1 and factorial zero. Factorial zero will be known when the procedure calls itself one more time. So the entire procedure terminates after multiplying 3 by 2 by 1 by 1, or 6, which is our answer...

—◉→

BASIC VS LOGO

Input and Output

Applesoft has "INPUT" and "GET"; these correspond to "REQUEST" and "READCHARACTER" in Logo. "READCHARACTER" can also be abbreviated to "RC." Applesoft can check whether the person running the program has typed something at the keyboard by testing whether the value of PEEK (-16384) is greater than 127. Logo does something similar by using the function called "RC?." This is a Boolean function, which means it returns a value of either True or False.

Both Logo and Applesoft use the "PRINT" statement, but the syntax differs. In Applesoft, you can print any number of items by separating them with commas (or semi-colons). Logo's PRINT defaults to a list of 2 or fewer items, and in order to print more than that, you enclose the entire statement in parentheses. Thus, we have the following example:

BASIC:
```
PRINT X,Y,Z
```

Logo:
```
(PRINT :X :Y :Z)
```

Printing of strings is different, too. Here is an example:

BASIC:
```
PRINT "WHAT NUMBER DO
YOU WANT TO COMPUTE THE
FACTORIAL OF?"
```

Logo:
```
PRINT [WHAT NUMBER DO
YOU WANT TO COMPUTE THE
FACTORIAL OF?]
```

The Logo example prints one item, that item being called a "list." There are no strings in Logo. What Applesoft would call a string would be called a list of characters in Logo. Actually, to be more accurate, lists are made up of words, not characters. (Of course, a word may be a single character.) Lists are always enclosed by square brackets, and each member of the list (word) is separated from the next by a space.

All of the PRINT statements above, both BASIC and Logo, will result in each item being printed, followed by a carriage return. As you know, in BASIC you can cause several items to be printed on the same line by separating them with semi-colons instead of commas. In Logo, you have to do this with a different statement, which uses the keyword "PRINT1" instead of "PRINT." This, unfortunately, can be much less convenient, especially when you want to print items aligned in columns. There are no Logo equivalents of the BASIC functions POS, SPC, or TAB.

Logo has a statement CLEARTEXT which does the same thing as Applesoft's HOME. NORMAL, INVERSE, and FLASH are not available except by doing what corresponds to POKEs in Applesoft. The function CURSOR involves two inputs and takes the place of VTAB and HTAB. There is nothing corresponding to any of these Applesoft keywords: READ, DATA, SPEED=.

Strings versus Lists

The list-handling capability of Logo is based heavily on another language, called LISP, which stands for "List Processing Language." Logo doesn't have strings, per se; what Logo deals with in a similar manner are words, lists, and sentences.

A word is merely any number of characters, without any intervening spaces. A list, as described above, is a group of words enclosed by square brackets. (Incidentally, you probably know that shift-M is the way of typing a right square bracket on the Apple; Logo has redefined the up-arrow, or exponentiation sign, to be the left square bracket.) A sentence is made up of either words or lists.

There are built-in functions called WORD, LIST, and SENTENCE to create these various entities. There are predicates called WORD? and LIST? which each take one input and return the value True or False depending upon whether the input has the corresponding property.

There are also operations you can perform on words and lists that are somewhat similar to Applesoft's LEFT$ and RIGHT$. The function FIRST applied to a word will return the first character; when applied to a list, it will return the first word in the list. The function LAST is used to return the last character/word. Then there are two functions which are the compliments of these, called BUTFIRST and BUTLAST. These will return all except the first or last character/word, as the case may be. Finally, FPUT and LPUT are used to add a word as either the first or the last in a list.

Logo also has a function ASCII which is like Applesoft's ASC, as well as a function CHAR which is like Applesoft's CHR$. There are no equivalents of MID$ or LEN.

By the way, words, lists, etc., are handled much like Applesoft's strings in another way: They involve the creation of "garbage", or unassigned values. So there is a statement ".GCOLL" to do the sort of "garbage collection" that Applesoft does with its function FRE.

Machine Language

There is a ".CALL" statement in Logo which corresponds to Applesoft's CALL. ".EXAMINE" corresponds to PEEK, and ".DEPOSIT" corresponds to POKE. There is nothing like Applesoft's USR.

Graphics

One of the real powers of Logo lies in its graphics capabilities. It's all done in High-Res, not Low-Res. The plotting of points and lines can be done either with x and y coordinates specified in a Cartesian coordinate system or by means of what is called "turtle geometry." I'll explain what I mean by these two concepts.

First let's examine the Cartesian system. This is where the x-coordinate is a horizontal component varying between -140 and +140, and the y-coordinate is the vertical component varying between -120 and +120. You can find the current x and y values associated with "the turtle" (the point currently being plotted or moved around), by using the functions XCOR and YCOR. You can set these values by the statements SETX, SETY, or SETXY.

However, the usual way of drawing in Logo is to use "turtle geometry." Here, there is an entity called the turtle which moves around over varying distances and with varying headings in order to plot lines.

You first call up the graphics capability with the command DRAW. (Thus it corresponds roughly to Applesoft's HGR.) To exit graphics mode, the command is NODRAW.

Upon entering graphics mode, you are shown the normal Apple High-Res screen, with the top 20 text lines being used for graphics, and the bottom four for text. This is SPLITSCREEN mode. You can also use the entire screen for graphics with the command FULLSCREEN.

The main difference between what HGR in Applesoft does, and what DRAW in Logo does, is the placing of a small white triangle in the middle of the screen (at x=0 and y=0.) This is the "turtle." If you do not wish to see this turtle, you use the command HIDETURTLE. If you want to see a hidden turtle again, you use the command SHOWTURTLE. (Whether the turtle is hidden or not makes no difference as to its plotting capability. Any movement of the turtle will cause a line to be plotted, unless you use the command PENUP. You might interpret this as meaning "raise the plotting pen." The converse of PENUP is PENDOWN.)

There are six "pen colors" you can use to plot on the screen. You set them with the PENCOLOR command; which takes a number between 0 and 5 as an input to express colors that range over black, white, green, violet, orange, and blue. Each of these colors can also be used as the background color with the BACKGROUND command.

When graphics is first entered, the turtle in the middle of the screen is "facing north" (i.e., up.) Its heading is defined to be zero at this time. The heading increases clockwise, so that a heading of 180 is "facing south." You can set the heading explicitly with the SETHEADING command, or you can modify it in terms of relative changes to its current heading, which are expressed as degrees rotated to the RIGHT or to the LEFT. Thus, a command of RIGHT 90 as the first one given to the turtle will cause it to be "facing east," while LEFT 90 will cause it to be "facing west." Of course, other values can be used with LEFT and RIGHT to rotate the turtle to face in any direction of the compass.

Motion of the turtle can be specified in relative terms, rather than Cartesian terms. The turtle moves either FORWARD (in the direction it is facing) or BACK (the opposite direction.) Each of these commands will take one input, this being the number of "units" of movement. (Remember, there are 280 units across the screen horizontally, and 240 vertically.)

I suppose this is called "turtle geometry" because young children being taught Logo graphics are encouraged to think of a turtle moving around to create whatever pictures they want to draw. In order to become really proficient at using this type of graphics, the children even (mentally) "become" the turtle and imagine themselves turning right or left and moving back and forth.

Here is an example of a simple procedure to draw a rectangle (box) which is N units wide by M units high:

```
TO BOX :N :M
   SETHEADING 0
   REPEAT 2 [FORWARD
   :M RIGHT 90 FORWARD
   :N RIGHT 90]
END
```

Once you've gone to the trouble of creating a graphics picture, you can save it to disk with the SAVEPICT command, and then you can get it back from the disk with the READPICT command.

If you create a real mess on the screen, and want to start all over again, you can clear the screen and place the turtle back in the center with the command HOME.

Miscellaneous topics

It is not my purpose to teach you Logo in this article, lengthy though it is. But I will end this discussion by briefly contrasting some of the other commands of Applesoft and Logo.

The command PRINTOUT is roughly equivalent to Applesoft's LIST statement. You can use it to display to the screen an entire procedure, or a list of all the names of the procedures you've currently defined.

ERASE is used to delete an entire procedure. (Remember, there are no line numbers to DELETE, as in Applesoft.) ERASE ALL corresponds to NEW.

SAVE does a writing of the current "work space" with all its defined procedures and names to disk. READ does a "LOAD." Other DOS commands can be done by following the word DOS with a list made up of the words in the command.

Summary

This has been just a brief introductory tour of Logo. If you find this language as intriguing as I did, I would suggest you rush down to your local computer store and purchase a copy of Terrapin Logo, or Krell Logo, or Apple Logo. The first two are nearly identical; Byte magazine recommends them over Apple Logo because the latter has no capability for inserting REMarks in programs. With the other varieties of Logo, you substitute the semicolon for BASIC's REM keyword. However, Apple Logo has a number of built-in capabilities the other two do not have. Ask your dealer, or read the Byte article....

THOUSANDS OF INNOVATIVE APPLE-COMPATIBLE PRODUCTS ARE GATHERING AT

Friday-Sunday
October 28-30, 1983

Moscone Center
10:30AM to 5:30PM daily

The Show features hundreds of exhibits of the newest, state-of-the-art products for the Apple. You can see and try out software for every conceivable application—from arcade games to investment programs, music to machine language, teaching systems to accounting packages, word processors to graphics processors. You can sample hundreds of different peripherals, including printers, hard disks, modems, memory cards, video displays, and synthesizers, plus accessories, publications and invaluable support services.

Applefest is **the** place to view the most technologically advanced products for the Apple.

At Applefest you can try out and compare hundreds of products in an exciting and information-filled environment. You can learn more in two days than you could in months of visiting computer stores and reading trade journals.

And, best of all, everything on display at Applefest is for sale at special show prices, so you can save hundreds—even thousands—of dollars by making your purchases at the show.

Admission Prices
One-day Exhibits Ticket $10.00
Three-day Exhibits Ticket $25.00

Additional Information
To receive more information about attending or exhibiting at Applefest call or write Northeast Expositions, Inc., 822 Boylston Street, Chestnut Hill, Mass. 02167. Tel: 800-841-7000 or 617-739-2000 (within Massachusetts).

Apple and the Apple logo are registered trademarks and service marks of Apple Computer, Inc.

Produced by Northeast Expositions, Inc. nationwide producers of PC'83, CP/M'83, the National Computer Shows and Softcon.

APPLEFEST TICKET ORDER FORM

☐ Enclosed is my payment for _____ Three-Day Exhibits ticket(s) quantity at $25 each, which admits the attendee to the Show on Friday, Saturday and Sunday.

☐ Enclosed is my payment for _____ One-Day Exhibits ticket(s) quantity at $10 each.

NAME _____

COMPANY (IF ANY) _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE ( ) _____ EXT. _____

Mail to National Computer Shows/Applefest, 822 Boylston Street, Chestnut Hill, Mass. 02167. Checks only. No charge cards.

Circle 25 on Reader Service Card

A Call —A.P.P.L.E. LANGUAGE REVIEW:

CEEMAC

A Visual Composition System

James L. Hockenhull/reviewer

CEEMAC *$75.00*
The Visual Composition System
by Brooke W. Boering

Vagabondo Enterprises
1300 E. Algonquin - 3G
Schaumburg, IL 60195
Not copy protected

Introduction

"CEEMAC," proclaims the cover of the 52-page manual, "is the language of Fire Organ." In case you've been living in a cave for the past year or so, "Fire Organ" is a demonstration disk of high-resolution (High-Res) art programs which has popped up at most Apple dealers. The disk is remarkable for its chosen method of distribution — free for the copying! — and for the speed at which the programs produce animated images on the video screen. Those who have dipped below the surface of Fire Organ to BRUN the several commentaries will have discovered that the graphic routines were all written in a language called Ceemac, the brain child of Mr. Brooke Boering, who *is* Vagabondo Enterprises. They will also have gotten a taste of Mr. Boering's enthusiasm, his philosophy, his idiosyncracies, and his sometimes unorthodox spelling.

Unlike most computer products, Ceemac and its documentation are heavily stamped with the personality of its author. In purchasing the package, one feels like one is supporting a crusade for a certain artistic viewpoint. An attack on Ceemac seems almost like an attack on Brooke Boering himself. And, writing a review is more like writing a piece of art criticism — but here goes!

What is Ceemac?

Pascal and BASIC strive to be all things to all people; Ceemac is a programming language which has only one purpose: to generate abstract, animated video graphics. It is designed to combine the programming ease of a high-level language with the running speed of machine code. Achieving such a goal will always involve compromises. Here they take the form of some rather stringent restrictions on the programmer.

At present Ceemac requires an Apple][or][+, 48K of programmable memory (RAM), one disk drive, and DOS 3.3. However, it strives toward "device independence," which means that features specifically tied to the Apple are held to a minimum. The Ceemac screen, for example, is 256 X 256, with the origin (point 0,0) in the lower left-hand corner, even though the Apple's screen is 280 X 192, origin at the upper left. The greater the degree of device independence the easier it is to modify the software for another computing system. Ideally, any Ceemac program will run on any future Ceemac implementation.

Ceemac is partly a block-structured language, similar in ways to Pascal: control structures such as loops are delimited by statements marking their beginnings and endings, and there are no line numbers. The good old GOTO and its target labels are still there, however, and, unlike Pascal, it is difficult, if not impossible, to program without using them. A number of commands are provided to perform specific tasks. For some reason, perhaps Mr. Boering's extensive experience as an assembly-language programmer, they are called "macros." The user cannot write new macros; subprograms are handled as BASIC-like subroutines, invoked by a GOSUB statement. As in BASIC, values are fed to the subroutines by assigning them to variables before the invocation or call. All variables are GLOBAL; they can be accessed by the main routine or any subprogram.

So far, so good: we've described a language that's a little bit like BASIC and a little bit like Pascal. But now Ceemac begins to veer away from ordinary languages, due in part to the necessary speed-vs-freedom trade-offs, in part to Brooke Boering's ideas about making art with a computer.

Variables

All Ceemac variables are predefined. The programmer cannot rename them, change their use, or add any more to the pile.

Let that soak in for a while.

Most of the variables are dedicated to specific uses by the system, uses which are usually pretty obvious. An assignment to the variable COLOR, for instance, will set the current color accordingly; variables X0.. X3 and Y0..Y3 specify X and Y coordinates for various uses. Interestingly, coordinate pairs can be treated as "pseudo-variables" and assigned values with a single statement:

```
XY1 = $80 ; 0
```

for example, will set X1 to $80 and Y1 to $0.

The "pool variables," P1..PF are set to random values by the macro NUPOOL and retain those values until NUPOOL is invoked again. "Finger variables" are those whose values are set by game paddles, buttons, or keys, allowing user interaction at run time. (Such interaction is optional; Ceemac is not merely another "etch-a-sketch" program.)

The entities RND1..RND7, although called "variables," are actually subprograms which, when invoked by name, return a single value. That is, they are FUNCTIONS. Every time a program states:

```
X1 = RND7
```

the variable X1 will be assigned a random value within a certain range. Similarly, "table variables" fetch and return values from internal look-up tables (lists). For instance, NXTCOL picks the next number from a table of colors; the assignment:

```
COLOR = NXTCOL
```

will advance the current color through the built-in color table, taking one step each time it occurs in the program.

```
COLOR = RNDCOL
```

will assign to COLOR a value selected at random from the table. NXTPRM and RNDPRM will do the same with a table of prime numbers. Four other tables are provided for indexed reference by the user. The first two, LIST1..LIST2, contain sine-wave values. LIST3 and LIST4 are for general programming use.

There are a number of other special-purpose variables and, fortunately, 15 "free" variables, V1..VF, which are at the user's disposal. All are single-byte locations, capable of holding 256 different values ($0..$FF). Ceemac, by the way, is most comfortable when dealing with hexadecimal notation, although it can deal with decimal numbers and will convert an entire program from one base to another automatically!

Control Statements:

Control of program flow is governed by the status of "C-EXPRSNS" (conditional expressions) which come in two forms: those operating on two values and those requiring only one. ("Binary" and "unary," respectively, in computer parlance.) Such expressions are built of "conjunctives," values, and operators, which sounds more confusing than it really is. Typical conditional expressions are:

```
WHILE V1 > 0
UNTIL X1 = X2
IF RND6 ODD
UNLESS PDL0 CHANGE.
```

Testable conditions like ON, OFF, ODD, EVEN, PLUS, MINUS, and CHANGE allow a great deal of flexibility in designing programs.

GOTO and GOSUB need no introduction. They differ in Ceemac from their BASIC counterparts in that they use labels (called "SYMS") to mark their destinations, since there are no line numbers. The GOTO can be made conditional by including a c-exprsn:

```
GOTO 1 IF V2 = V4.
```

The general-purpose looping structure is bounded by the word DO at the beginning and AGAIN at the end. A conditional expression may follow either or both of these delimiters:

```
DO      <c-exprsn>
        <loop body>
AGAIN   <c-exprsn>.
```

If no conditional is present at either place, the loop goes on forever. The entire main program exists in an implied infinite loop, that is it will repeat itself until the plug is pulled unless the programmer specifies otherwise.

The indexed or counting loop begins with the word FOR followed by a single value which specifies the number of times the loop is to be repeated. It, too, is terminated by AGAIN:

```
FOR     <value>
        <loop body>
AGAIN.
```

There is no way to access the loop index.

Loops and subroutines can be exited prematurely with the EXIT statement, which may also include a conditional expression. SKIP will skip over the next statement; it, too, can (and usually does) include a c-expression. These control structures are quite powerful.

Ceemac incorporates the usual arithmetic operators (+, -, *, /) and, since it deals with single-byte quantities, it includes means to use either the high-byte or low-byte result of multiplication or division. The novel operator "+OR-" either adds or subtracts its operands — at random! The operator "%" adds until the sum is greater than 255 (overflow), then subtracts until the difference is less than 0 (underflow), at which point it begins adding again. This allows events on the screen to reverse their direction when reaching an edge, rather than wrapping around.

(As mentioned above, Ceemac deals with a "unit screen" considered to be 256 X 256 points. The unit screen is scaled or "mapped" onto Apple's 280 X 192 display. The proportion ("aspect ratio") of the mapped image can be distorted somewhat by the macro SETASP. Note that the dimensions of the unit screen allow both X and Y coordinates to be single-byte (0..255) values.)

Logic operators are AND, OR, and EOR (exclusive OR). In Ceemac jargon, these are lumped together with the arithmetic operators as "Math-A-Logical" operators.

Macros

Macros are Ceemac's commands. Some handle the basic procedures needed by any graphics system: CLEAR the screen, plot ADOT, draw a BLINE. (No, I don't know what the "B" stands for.) Others extend the graphic capabilities, allowing outline boxes and filled rectangles to be drawn, as well as the graceful compound curves called SPLINES or "Jordan Curves." A proprietary shape system, different from Apple's is also incorporated into the language.

Macros take up to two arguments whose effects differ from command to command. In many cases the second argument selects from 12 different orders of symmetry. Yes, Ceemac will automatically rotate and reflect points, lines, curves, and boxes, one of its many fascinating properties — and, since not everyone would agree on the value of axial symmetry, a good example of how artistic preferences can be built into a language.

Other Stuff

A review isn't the place to itemize every feature of something as complex as a language package. I will mention that Ceemac can generate tones through the Apple's speaker, can read values from the cassette-in jack, making possible an admittedly crude sort of interaction to musical accompaniment, and can doubtlessly do things I haven't discovered yet.

Programming in Ceemac

Programming is done through an excellent screen-oriented text editor. ("Screen-oriented" means that there are no line numbers; the cursor is simply moved around through the text of the program.) As in interpreted BASIC, the editor does some pre-compiling so that syntax errors are caught and reported immediately. Ceemac is *very* friendly and helpful, formatting the program listing to reflect its structure (nested loops, for example, are automatically indented to the proper levels), inserting square brackets around macro argument lists, and even supplying default values for things the user may leave out. It definitely wants to hold your hand.

Typing [Ctrl-A] at any time will exit the editor and execute the program being worked on. Another [Ctrl -A] stops execution and returns to the editor, with the cursor positioned right where you left it. Interaction like this allows programs to be developed intuitively at the keyboard by what Mr. Boering calls the "trial and wow!" method. To encourage such an approach, the system is robust and idiot-proof; this particular idiot has not caused a system crash yet. Actually, it's a good thing that the seat-of-the-pants attack is supported because fooling around is really about the only way to get a grasp on Ceemac, thanks to the opaque documentation.

The Manual

Notice to software documentors everywhere: One well-chosen example is worth paragraphs of technical description. Omitting tutorials is not a mark of sophistication — quite the opposite. In doing so you do a disservice to your own product; you cut your own throat. Period. Amen.

The Ceemac manual is comprehensive, is detailed, is complete (well, almost). It is emphatically *not* tutorial; there are virtually no examples of what is being discussed. (Although source files of the Fire Organ programs are included on the disk.) Its organization is weird. The writer has the annoying habit of introducing a term, then telling us that it will be explained somewhere else, seldom mentioning where. And if this is not enough, the booklet is laden with Mr. Boering's own idiosyncratic terminology: programs are "scores," users are "composers," certain keys are "drums." We have "math-a-logical" expressions, "finger variables," "forces," and "ghost-points." (Many who march to a different drummer share this passion for coining their own phrases: the designer and utopian thinker Bucky Fuller, comes to mind, or the late, eccentric composer Harry Partch.)

In short, and to be as kind as possible, the manual is confusing. That this is not merely personal opinion is borne out by three facts:

1) Mr. Boering himself refers to Ceemac as "cryptic" and periodically tells the reader to fool around on the keyboard if the manual gets too heavy.
2) Issue #0 of the independently published *Ceemac Newsletter* talks of "rumblings from folks who are finding Ceemac somewhat mind-boggling."
3) In the ultimate admission that the manual is not doing what it should, Boering has finally issued (as the draft of this review was being typed) an eleven-page pamphlet, the *Ceemac Tutorial.* Although it does not touch on many of the language's more unusual features, it goes a long way toward preparing users for sallies into the dense thickets of the manual. He is to be congratulated on the tutorial. Presumably it will be included in future documentation.

The Big Picture

Ceemac is wonderful fun. It should be the delight of kaleidoscope fanciers. With the proper documentation, kids would love it. It has the potential for great entertainment value. But Ceemac is being proferred as something beyond mere entertainment. It is "a composing tool in which artists of this new medium can record their visual 'songs.' " Can one really make art with Ceemac?

That question cannot be answered without a clear definition of *Art,* a definition which has eluded philosophers for many, many centuries. Brooke Boering's ideas about the nature of art, ideas which are cemented into the very foundations of the Ceemac language, exemplify what I call the Numerological Aesthetic. (Numerology refers to a system of occult beliefs centering around the magical significance of numbers.) This attitude looks toward music as the great universal art-form, evidenced here by Boering's repeated use of musical terminology.

Another example of this outlook can be seen in John Whitney's book *Digital Harmony.*) The thinking goes: the harmonies and dissonances of musical sounds and structures are based on certain numerical relationships; if we apply similar relationships to visual forms we'll come up with a universally satisfying art. Many who hold this view find spiritual support in the paintings and writings of the 20th-century abstractionist Wassily Kandinsky. True to form, Mr. Boering prefaces the Ceemac manual with a page of Kandinsky quotes, one of which says, "The final abstract expression of every art is number." The presence of all those numbers appears to give art a good solid basis in rationality. But I suggest that the essence of this attitude is a kind of Pythagorean mysticism, a blind faith that quasi-musical number play will be instantly recognizable and artistically pleasing.

Micro Match™
SAVES MONEY
Don't Let Programs Gather Dust!
TRADE Original SOFTWARE with Documentation for
ALL MICROS . . . ALL PROGRAMS
WE MATCH TRADERS
for only $5.00
GUARANTEED SATISFACTION
FOR UNLIMITED TRADING:
Ask about SOFTRADERS Worldwide Trading Network
- ***Quarterly Directories***
- ***Monthly Newsletter***
- ***Cash Bonuses • Much more***

Send your system description, titles of all Software wanted, titles of all software to trade and S.A.S.E. Send no money.

SOFTRADERS™ International
1610 Shomaker Drive
Murphysboro, IL 62966

Circle 30 on Reader Service Card

CEEMAC

Personally, I question the Numerological attitude. I haven't seen convincing evidence that a simplistic one-to-one mapping of phenomena of the ear onto phenomena of the eye creates much beyond decoration. I feel that computer art will prove to be more closely akin to the field of artificial intelligence than to music. And to tackle the *enormous* problems of intelligence and the modeling of behavior, one needs the scope of a powerful, general-purpose language. Ceemac makes certain kinds of things very easy and other kinds of things very difficult, if not impossible. Through its design, Brooke Boering leads the user by the hand down a narrow, though scenic, road through the land of computer visuals. But, to paraphrase novelist Margaret Atwood, a road is somebody's way of making you go where they want you to go.

Summary

Ceemac is a unique tool for the creation of rapidly-moving High-Res images. Certain of its author's biases are reflected in features such as those which support symmetry and sequences of prime numbers and sine values, features which facilitate, hence encourage, specific approaches to art-making. While some may feel restricted, others will enjoy the ease with which impressive "compositions" can be created. The language is structured (a welcome relief from BASIC) and interactive, the system is friendly and robust. As a bonus, the purchaser indirectly meets a very interesting person, Mr. Brooke Boering. Ceemac's major weakness is its documentation, a situation which shows promise of improving steadily.

A fine way for those with some programming experience to begin exploring the creative potential of the computer.

Why Your Apple Needs DAVID-DOS™

Your Apple is easier to use and runs much faster with this new, licensed by Apple, DOS 3.3 update.

DAVID-DOS (rhymes with moss), is rated AA, (top of its class), by the highly respected software review magazine PEELINGS II.

Novices and Pros

DAVID-DOS is licensed by programmers for inclusion in the software they sell. 30 out of 60 bought it on the spot when DAVID-DOS was demonstrated at the Original Apple Corps, UCLA. DAVID-DOS receives applause from Apple novices. That's why you need DAVID-DOS.

What's Going On

Every Apple II/IIe boots up with DOS 3.3 software.* DOS 3.3 is right there on the first three tracks of all your program disks. Your Apple is supervised constantly by DOS 3.3.

Easier and 5 Times Faster

Our Installation Program goes in and makes specific permanant improvements to DOS 3.3 on all your disks in a few seconds. (The programs on your disks are not touched). From then on your programs load up to 5 times faster. And you have new features that make your Apple easier to use.

100 Sectors in 7 Seconds

Speed Load Applesoft, Integer & Binary 100 sector programs in 7 seconds. Tload Random and Sequential Text Files at the same speed with a simple command.

Automatic Card Support

Automatically supports an Integer or Applesoft Rom Card in Any slot (without configuring). Single key stroke Catalog and Catalog Abort. Shows Free Sectors Left on each Catalog.

Nine New DOS Commands

1. TLOAD speeds loads all Text Files.
2. TLIST lists all Random/Sequential Text Files.
3. DUMP Binary/Ascii to screen or printer.
4. DISA dissembles Binary to screen/printer.
5. AL prints program Address & Length.
6. / Single keystroke, second Catalog command.
7. HIDOS moves DOS to the Language Card.
8. FIND hex group in 64K memory in 3 seconds.
9. DATE prints out. Use with Mtn. clock card. Commands 8 & 9 in Hidos only.

10K More Memory

These nine commands operate identical to existing DOS commands. Use A or A$ for address and L or L$ for length. Enter new commands on the Keyboard and use them in Basic programs with the familiar D$. Use HIDOS in Hello program for turnkey startup, adding 10K free memory.

Variable Speed Scrolling

TLIST, DUMP, and DISA Features are:
1. Print with PR#.
2. Key operated variable speed control.
3. Instant pause with Space Bar.
4. Exit anytime with Return Key.

Compatible

All DOS entry addresses have been preserved. DOS is Same length and compatible with most software. David-Dos is copyable and creates fully copyable updated disks. Init areas were used for David-Dos. Works with all Apple IIs including IIe 80 Col. Requires 48K. Complete documentation for screen or printing and many utilities are on the disk.

California residents add 6½% sales tax. **$39.95** Foreign add $2.00.

*Exceptions are Pascal & CPM Operating Systems.

Apple II and Applesoft are trademarks of Apple Computer inc.

DAVID DATA™
To Order: Send Check To
12021 WILSHIRE BLVD., SUITE 212A
LOS ANGELES, CA 90025 (213) 478-7865

Circle 16 on Reader Service Card

A Call —A.P.P.L.E. SOFTWARE REVIEW:

THE GRAPHICS MAGICIAN

Harvey Gratt

The Graphics Magician $59.95
Penguin Software
830 4th Avenue
Geneva, IL 60134
Unprotected
Updates from old to new version:
$15.00 with original disk and manual.

THE Graphics Magician package by Penguin Software is one of the most useful (and just plain fun!) graphics utilities on the market today. Not that it is without fault, the documentation is sketchy, somewhat difficult to follow for the beginner and does not provide enough examples or meaningful diagrams which would greatly enhance comprehension. All of this aside, the program delivers everything that it's advertised to do, with its potential applications limited only by the users' imagination.

The software is resident on one full unprotected diskette and consists of the main three sets of programs (Animation System, Picture/Object System and Super Shape System) along with several utility programs and demos.

The animation routines consist of several editors and associated machine language programs designed to simplify the development and incorporation of two dimensional animation into the users' programs. A Shape Editor allows one to easily design a given object (up to 32 objects may be incorporated in a program simultaneously) with colors for animating within one's programs. The Path Editor is invoked to define a given path around the screen in which the object is to be moved. Finally, the Animation Editor allows the linking of up to 32 different objects and associated paths under one of four different animation types. Machine language routines are next linked with the assembled data, resulting in a block of code and data which can be loaded into the users' programs and called for performing the desired animation.

For the adventure game or educational software designer (and others!) the package includes a high-resolution Picture/Object System that lets the user draw, using any of 108 colors, pictures with object overlays. These may then be stored on disk in a compressed (instruction oriented) form and later quickly recalled and displayed from one's own programs. The compressed mode of storage, consisting essentially of the instructions used to draw the picture (to be subsequently linked to a picture draw routine), allows for the packing of literally hundreds of High-Res pictures on a single disk.

Several utilities and options are also included:

1) A disk access option provides for the owners of two drive systems to designate a "Master Disk" and a "Data Disk" drive specification.
2) A Binary File Transfer Utility program allows for the easy transfer of files from one disk to another. Pertinent information (starting address and length) is displayed for future reference.
3) A faster (20%) picture draw routine (PICDRAWF) is also included. Its use is identical to the standard version but does occupy 512 more bytes of memory.
4) A collision detection subroutine is provided and can be loaded and accessed by the users programs.
5) A "Shape Start" routine is included for converting existing Applesoft shapes to the pre-shifted shape used by the animation routines.
6) An option to save pictures in the standard (Penguin) screen format (.PIC) is also available. This allows one to use pictures generated by the Picture/Object Editor as backgrounds for the animation system as well as allowing compatibility with all other Penguin graphics software.
7) Finally, several demo programs and an "Animated Alphabet" are included for use as is or for easy modification by the Animation System.

Graphics design is implemented by menu-driven editors with the final products being linked to the appropriate machine language routines for incorporation in the users' programs. The editors themselves are fairly well structured from a user's point of view as evidenced by the fact that it was easier to *use* the graphics utilities than to learn how to use them from the documentation. Some observations concerning the various major subsystems (along with the few minor idiosyncrasies encountered) are now discussed.

Animation System

This is probably the most impressive set of routines contained in the Graphics Magician. It is geared for those individuals who desire to incorporate the most professional looking animation effects in their software. The routines are written around the technique of "pre-shifted" shapes and allow the option of using any of four different animation types. Pre-shifted shapes entail the simultaneous creation of seven "bit-shifted" replicas of the desired shape which are then stored in memory. This allows for smooth motion of shapes when their movement consists of horizontal changes which are less then byte increments in length. The animation types consist of so called "block moves" (with and without a High-Res page 2 background option), a "draw with background save" and an "XDRAW" option. The distinguishing feature of the block move options is that an erase cycle is not needed for the animation effects. Borders around the shapes essentially "erase" the remnants of the object as it moves along its path. The end result is a fast, smooth and flicker-free animation. The latter two modes, while possessing some flicker, allow for more complete collision detection capabilities.

Design of an animated object begins with its creation via the Shape Editor. The routine is entered directly through the main (boot-up) menu.

The user is presented with a High-Res display showing seven orange lines and associated cursors corresponding to the pre-shifted shapes. The text window present on the bottom of the screen offers a list of single letter commands and status information which are used to construct the desired shape.

Cursor movement is keyboard controlled through the familiar I,J,K,M keypad arrangement. Plotting is accomplished by single letter commands which act as a toggle for plotting (or erasing) single or multiple points at the users discretion. In general, each cursor function operates on all seven pre-shifted shapes unless they are specifically toggled off. By the selective turning on and off of the seven shapes, each can be made slightly different, allowing a wide range of animation effects within the shape itself during horizontal movement. The net effect is very impressive. An option for plotting even or odd points allows either for reproduction of the standard Applesoft colors.

After the shape has been designed the user indicates that he/she is finished by typing a "D". A secondary set of single letter commands are then at his/her disposal for (S)aving, (C)ompiling or test (A)nimating. Direct entry into this level of the Shape Editor also allows one to (L)oad a previously saved shape for further development.

While all of this may sound somewhat complicated, it is quite straightforward to learn in a hands on (computer on?) environment. The single letter command structure is conducive to rapid editing and the design of the shapes can be easily (although sometimes tediously for complicated shapes) performed once one gets the hang of the overall operation. In fact, direct experimentation is definitely the way to go, since the documentation provided is somewhat sketchy in explaining the details of what is taking place.

A few minor annoyances were encountered. While test animating a shape, an inadvertent keypress (a (L)oad or (C)ompile instead of the spacebar for exiting) throws one into that mode without a way to conveniently exit. Also missed was the ability to catalog a disk from the Shape Editor without returning to the main menu.

Once a shape has been designed and saved to disk, the Path Editor is then entered. Again under keyboard control, an arbitrary path (up to three separate paths may be combined to subsequently describe an objects motion) may be constructed with motion possible in any of eight primary directions (others are possible by combining the primary moves) and of varying lengths. Path construction can be accomplished quite easily with mistakes being simply corrected by (D)eleting the erroneous move. Upon completion, the desired path may then be saved to disk and the user returned to the main menu.

The same minor concerns are present here as were in the Shape Editor. One cannot catalog a disk, or worse, load a file for further development (this is an oversight). Again, one may find himself in a command mode (i.e. (S)ave) with no convenient way to abort it.

After one or more shapes and associated paths have been designed, the Animation Editor is then used to select an animation type and link the desired shapes and paths together, forming up to 32 different objects capable of concurrent animation. The disk resident machine language animation routines may then be linked with the object data set to form an animation file which can then be called from the users programs in order to provide the desired animation effects.

The Animation Editor is also menu driven, allowing one to selectively load the desired shapes and paths for assembling into objects. Cataloging of the users data disk is provided for in this routine. The user then selects the (N)ew object option and is provided with prompts which allow him to assemble any given shape with up to three different paths to form an animation object. The starting location and position within the path for the assembled object are also provided for. All object data may be modified at will by a convenient (E)dit mode option.

With a minimum of experimentation, the assembling and linking of machine language routines and object data into an animation file is easily accomplished. Somewhat more difficult to assimilate, however, is the technique for performing animation from within ones' programs. The primary manual has a brief and sketchy discussion, followed by a description of the pertinent data sets and associated tables that the user may access. If it were not for the inclusion of a programming tutorial (apparently not included with early versions of the Graphics Magician) the beginner (and not so beginner) would be hard pressed to make use of the inherent power and flexibility that Penguin has provided. In short, the primary documentation, while not terribly bad, is inconsistent in its presentation, ranging from shallow non-informative passages to rather technically dense material without providing the reader an easy learning environment. I do feel, however, that there is an opportunity here for a companion "textbook" on graphics software in general; more about this later.

All in all, the Animation System is well done, extremely powerful and quite flexible. Fortunately, the inclusion of a programming tutorial along with the inherent ease of use of the software itself more than compensates for the slightly subpar primary documentation. It should be emphasized that the documentation is not bad but is simply not up to the level of the software.

Picture/Object System

This part of the Graphics Magician package allows one to draw highly detailed High-Res pictures and objects which can later be saved in a highly compressed form, thus allowing storage of hundreds of picture files on a single disk. The user has at his disposal 108 colors and eight different brushes for performing fine detail work. Using either paddles or joystick, pictures may be laid out on a white background and colored using a very fast fill routine which colors the users selected enclosed areas. The fundamental mode of operation consists of locating a cursor at a desired starting point and "setting" it via button 1. The cursor is then positioned at the end of the desired line segment and button 0 is depressed thereby drawing the line.

The whole operation is easy to learn and a lot of fun to work with. Here, the manuals do a good job of explaining all that is necessary to get one started.

As with the animation routines, a single letter command set is displayed in the text window on the bottom of the High-Res screen. The user is given rather complete control and editing flexibility for reducing the tedious work to a minimum. Even provided are commands to select joystick orientation and to home in on the current drawing area (cursor) with a desensitized joystick (paddle) mode — almost a necessity due to the inherent instability of many of these devices. Upon completion of the picture or object, the user then saves his work to disk for later inclusion in his programs. Both the regular manual and programming tutorial then describe how to load the desired picture draw routine (PICDRAW or the faster PICDRAWF) and subsequently load and link the desired pictures for use in ones programs. If so desired, the user may also combine many individual pictures into one large binary file, thereby reducing disk accesses during the application programs execution.

Some comments concerning overall impressions are in order:

1) This system uses either joysticks or paddles only — this is no match for a graphics tablet, and things can get tedious in a hurry. It almost seems as though a keyboard control option would help in many situations, such as drawing long line segments where (x,y) coordinates can be explicitly specified.
2) As in some of the animation editors, one can get into a mode (i.e. (L)oad, (S)ave) with no easy way to abort if so desired.
3) The compressed picture format is nice! The ability to store large numbers of pictures on a single disk will be truly appreciated by the serious user.
4) The color fill routines are fast and generally can perform the desired action on one attempt. Irregular shapes will usually require several passes to completely color them.
5) The .PIC (S)ave option is non-operative in the versions of the program I have looked at (including the latest). Examination of the editor source code (written in BASIC) shows the necessary code but with no way to access it. This is something that Penguin needs to patch.
6) It would be nice to have several "canned" shapes included. This might allow the user to quickly draw shapes such as circles, squares, triangles etc. by merely specify a start location and a few parameters (i.e. center and radius for a circle).
7) The inclusion of some text writing ability on a picture would also be a nice enhancement.

Needless to say, these points are minor in nature and to some degree simply a "wish list" (you just can't please everyone!). All in all, the Graphics Magician is a very neat and fun system to work with — it was all I could do to get my kids away from it so I could play with it!

Summary

So you ask, what's the bottom line? The Graphics Magician certainly delivers everything it's advertised to do. Its power and flexibility makes it the kind of tool that frees the user from the tedium of implementing animation effects and picture drawing per se, thereby allowing concentration on the overall structure of his programs instead. The resulting time-savings in coding and debugging are obviously great. Criticism of the package is essentially trivial — ranging from the previously mentioned "wish list" additions to a somewhat more desirable improvement in documentation. As I've indicated previously, it seems that there would be a market for a rather complete "textbook" which would allow the reader to learn the "state of the art" techniques now employed by the professional game writers. What better vehicle for introducing computer science concepts (flowcharting, assembly language, data structures, graphics techniques, etc.) than the design of an arcade type game. I certainly feel that this would be a welcome addition for the serious computer user. In the meantime, my kids will never forgive me if I don't purchase a copy of this program!

Note from Penguin: The newest version of Graphics Magician was released in the first week of July. It includes several of the items on the "wish list", as well as an all-new manual (at last!) A lot of new editing features have been added, and all the machine language routines are now even faster and more compact. To update from the old version, current owners should return their original disk and manual with $15 to Penguin.

BYTES & PIECES

Extra Special!
Multiplan
[CP/M or DOS]
$192.19
$1.00 credit for all phone orders over $100.00

	LIST	OURS
Bag of Tricks	39.95	29.56
Double Take	34.95	25.86
Graforth	75.00	55.49
Graphics Magician	59.95	44.36
Harcourt Brace SAT	79.95	59.16
Home Accountant	74.95	55.46
Krell Logo	89.95	76.46
Krell SAT	299.95	254.96
Micro Cookbook	40.00	29.56
Nibbles Away II	69.95	51.76
Police Artist	34.95	25.86
SAM	124.95	92.49
16 K Card (Microsoft)	99.95	77.96
Super Fan II (w/zener ray)	109.00	80.66
Tasc Compiler	175.00	129.49
Witness	49.95	36.96
Word Handler	199.00	147.26
Zoom Grafix	49.95	36.96
SPECIALS		
Choplifter	34.95	24.86
Zaxxon	39.95	28.56
FCM	99.95	68.96
Format II	150.00	107.96
Wildcard	139.95	111.19
Replay II	130.00	107.19
Bank Street Writer	69.95	49.96
Sensible Speller	125.00	90.49

C.O.D. • Money Orders • Certified Checks • Personal Checks Allow 2 Weeks • N.Y.S. Res. Add Sales Tax
U.S.-Orders Under $150. Add $2.00 P&H
All Canadian, U.S. Funds $3.00 P&H
Foreign, Charges Only. Min. P&H $6.00
MasterCard
SEND FOR FREE PRICE LIST #908
Source TCP 637 • Compuserve 72135, 1710
BYTES & PIECES (516) 751-2535
P.O. Box 525 Dept C • East Setauket, N.Y. 11733
Circle 45 on Reader Service Card

Laing Electronics

10382 STANFORD AVE.
SUITE E-203
GARDEN GROVE, CA 92640
PHONE: (714) 534-4216

MICROSOFT 16K CARD
$69.00

★ HARDWARE FOR THE APPLE II ★

	LIST	YOUR COST
HAYES MICROMODEM II	379.00	258.00
HAYES SMARTMODEM 300 BAUD	289.00	199.00
HAYES SMARTMODEM 1200 BAUD	699.00	498.00
NOVATION APPLE CAT II	389.00	268.00
MICROSOFT Z-80 SOFTCARD CP/M	345.00	219.00
MICROSOFT PREMIUM PACKAGE (Z-80, VIDEX 80 COL., 16K CARD)	695.00	490.00
PCPI 6 Mhz APPLI-CARD (Z-80 CP/M)	375.00	279.00
VIDEX VIDEOTERM 80 COLUMN CARD (SOFTSWITCH, AND INV. CHAR. SET)	319.00	225.00
WIZARD 80 COLUMN CARD	249.00	149.00
WIZARD BUFFERED PINTER INTERFACE (16K BUFF + GRAPHICS)	179.00	129.00
(32K BUFF + GRAPHICS)	219.00	153.00
NEC HI-RES GREEN MONITOR (JB1201)	285.00	153.00
MICRO-SCI A2 APPLE COMPATIABLE DISK DRIVE W/O CONTROLLER	345.00	249.00
MICRO-SCI A2 DRIVE WITH CONTROLLER	445.00	299.00
OKIDATA MICROLINE 92/P PRINTER	699.00	498.00
PROWRITER 8510AP PRINTER	495.00	349.00
KENSINGTON SYSTEM SAVER FAN	89.95	65.00
GIBSON LIGHT PEN/LPS II (PENTRACK)	349.00	239.00
VISTA V1200 CART. DISK SYSTEM 6.2MB	1840.00	999.00
VISTA A-800 8" DISK CONTROLLER	379.00	249.00
VISTA SOLO 5¼" DISK DRIVE	299.00	195.00
VISTA SOLO 5¼" DRIVE W/CONTROLLER	379.00	245.00
VISTA QUARTET 5¼" DISK SYSTEM (2 THINLINE DRIVES, AND CONTROLLER)	849.00	573.00

★ VERBATIM DISKS ★

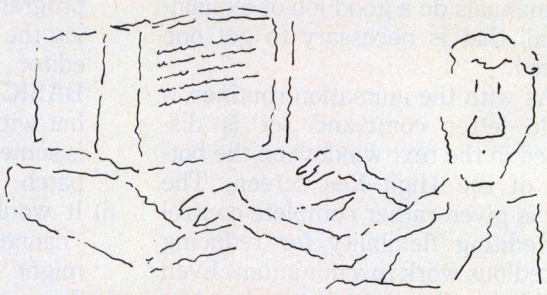
100 DATALIFE 5¼" (MD525-01-18158)	490.00	239.00
50 DATALIFE 5¼" (MD525-01-18158)	245.00	124.00

★ SOFTWARE FOR THE APPLE II ★

PFS: FILE (SOFTWARE PUB.)	125.00	76.00
PFS: REPORT (SOFTWARE PUB.)	125.00	76.00
PFS: GRAPH (SOFTWARE PUB.)	125.00	76.00
DB MASTER REG. VERSION (STONEWARE)	229.00	147.00
DB MASTER VERSION #4 (STONEWARE)	350.00	225.00
MULTIPLAN (MICROSOFT) CP/M OR DOS3.3	275.00	167.00
WORDSTAR (MICRO-PRO CP/M)	395.00	299.00
WORDSTAR WITH Z-CARD (PCPI APPLI-CARD)	495.00	359.00
SCREEN WRITER II (ON-LINE)	129.95	79.00
FORTRAN-80 (MICROSOFT CP/M)	195.00	125.00
ASCII EXPRESS PROFESSIONAL	129.95	85.00
Z-TERM PRO. (SOUTHWESTERN DATA)	149.95	98.00

If Personal Check, Allow
10 BANKING DAYS for CLEARANCE.
Prices and specifications subject to change without notice. Unless otherwise noted,
ALL PRICES are PLUS SHIPPING.
SHIPPING & HANDLING—ADD $5.00 UPS SURFACE (DISK) DRIVES $7.50—MONITORS $10.00—PRINTERS $15.00.
California residents add 6% Sales Tax.

Circle 20 on Reader Service Card



Get a Grip on Machine Language.

The Visible Computer: 6502

The transition from Basic to machine language is a big step. Sometimes too big.

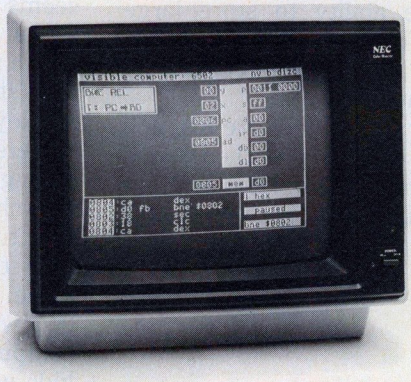
Now there's a way to cut the problem down to size.

The Visible Computer: 6502 is an integrated system for learning machine language programming. It features an animated simulation of the 6502 microprocessor that lets you see with your own eyes what machine language is all about. You'll see how instructions are executed, not just the result. There's a hand-holding mode that traps the kind of mistakes beginning machine language programmers are apt to make. And an expert mode, with more than twenty commands and functions that make it a powerful debugger for your machine language programs.

The 140 page manual is a hard-working tutorial on the fundamental concepts of machine language. You'll learn "hands-on-keyboard" as you work your way through thirty progressively more complex demonstration programs stored on disk.

The Visible Computer: 6502 is available for the Apple II Plus (or IIe) for $49.95. If your dealer doesn't have it, you can order directly from Software Masters, 3330 Hillcroft #BB, Houston, Texas 77057, or call (713) 266-5771. Please include $3.00 shipping. Bank cards accepted.

The Visible Computer lets you see into a 6502 as it executes programs



Software Masters™

Circle 38 on Reader Service Card

APPLESOFT GRAPHICS MOVER

Homer O. Porter

RECENTLY, after creating a high resolution picture, I decided that I wanted to move the picture up to center it on the screen. Assuming that someone had probably solved this problem, I looked for a program to move a high resolution picture. I could not find anything in my limited library (mostly *Call -A.P.P.L.E., Apple Orchard* and *Softalk,* with a few other commercial magazines). What I did find was an article (Ref. 1) on the layout of the high resolution screen. Therefore, I decided to write a program to move the picture. The program was written in Applesoft on a 48K Apple][Plus.

To understand the program, it is necessary to comprehend the layout of the high resolution screens. As most of you know, there are two high resolution screens, Page 1 located from 8192 ($2000) to 16383 ($3FFF) and Page 2 located from 16384 ($4000) to 24575 ($5FFF). For some undoubtedly good reason, the location on the screen and the location in memory are not sequentially related. The screen is divided into thirds. Within each third, there are eight sections, with each section containing eight lines and each line containing 40 bytes of memory. The only part of the screen that is in sequential order in memory is each line. Fortunately, there is a pattern.

By studying Reference 1, especially chart #4, I found the following pattern. The first memory location is 8192 (Page 1) or 16384 (Page 2). The first location of the next third of the screen is 8063 bytes less than the last location of the previous third of the screen. Within each third of the screen, the first location of each section is 7079 bytes less than the last location of the previous section. Finally, within each section, the first location of each line is 985 bytes more than the last location of the previous line. As mentioned earlier, each line is 40 bytes long, in sequential order. This pattern will be used to determine where each screen location is found in memory.

```
]LIST

100  REM

GRAPHICS MOVER  BY HOP PORTER

CALL -A.P.P.L.E. : SEPT. 1983

200  REM
SET UP - LOADS HI-RES  PICTURE ON
     PAGE 2

210  TEXT : HOME
220  VTAB 10: PRINT "GRAPHICS MOV
     ER MOVES A HI-RES PICTURE
          UP OR DOWN ON THE SCRE
     EN"
230  PRINT : HTAB 12: PRINT "BY H
     OP PORTER"
240  HTAB 13: PRINT "CALL -APPLE"

250  PRINT : PRINT : INPUT "NAME
     OF HI-RES PICTURE FILE? ";NA
     ME$
260 BEL$ =  CHR$ (7)
270 D$ =  CHR$ (13) +  CHR$ (4)
280  HGR2
290  PRINT D$;"BLOAD";NAME$;",A$4
     000": REM

300  REM
MOVE DATA - REARRANGES AND MOVES
PAGE 2 (16384 TO  24575) DATA
TO PAGE 3 (24576 TO 32255)
BYTE BY BYTE

330 PIC = 16384:TEMP = 24576
340  FOR SCREEN = 1 TO 3
350  FOR SECT = 1 TO 8
360  FOR LINE = 1 TO 8
370  FOR BITE = 1 TO 40
380  POKE TEMP, PEEK (PIC)
390 PIC = PIC + 1
400 TEMP = TEMP + 1
410  NEXT BITE
420 PIC = PIC + 984
430  NEXT LINE
440 PIC = PIC - 8064
450  NEXT SECT
460 PIC = PIC - 984
470  NEXT SCREEN: REM

480  REM
FINISHED WITH PICTURE

500  REM
WHAT NEXT? - MOVE THE  PICTURE,
SAVE THE PICTURE OR QUIT

530  TEXT : HOME
540  PRINT "MOVE PICTURE, SAVE IT
     , OR QUIT? (M/S/Q)";: GET A$
     : PRINT A$
550  ON A$ = "M" GOTO 630: ON A$ =
     "S" GOTO 1030: IF A$ = "Q" THEN
      PRINT "THE END": END : REM


600  REM
MOVE PICTURE UP OR DOWN

630  INPUT "HOW MANY LINES TO MOV
     E UP? ";X
640  POKE 49232,0: POKE 49237,0
650 PIC = 16384:TEMP = 24576 + (4
     0 * X)
660  IF  SGN (X) =  - 1 THEN  GOTO
     770: REM

690  REM
MOVE UP

700 COL =  PEEK (32255)
710  FOR I = 1 TO (40 * X)
720  POKE (32255 + I),COL
730  NEXT I: GOTO 810: REM

760  REM MOVE DOWN

770 COL =  PEEK (24576)
780  FOR I = (40 * X) TO  - 1
790  POKE (24576 + I),COL
```

APPLESOFT GRAPHICS MOVER

With that background, I was ready to determine an approach to moving the picture. After some consideration, I decided to collect the data from each memory location and store it in some other memory location in the order it was displayed on the screen, then change it, and return the changed data to the high resolution memory locations. I chose to store the data in memory above high resolution Page 2, starting at 24576 ($6000). Therefore, the program collects the high resolution data from each line starting at the top of the screen and moving down one line at a time. These values are stored in the sequence that they are read, giving me the picture stored in a block of memory starting in the upper left of the picture and going line by line down the picture, ending in the bottom right. Now, the picture can be moved up or down in this new location by simply adding or subtracting lines of the appropriate color. After doing this, the data is moved back to the high resolution memory.

Listing 1 is the final program. Lines 100 to 240 are the title. Lines 250 to 290 load the picture into Page 2 of the high resolution screen. I chose Page 2 because that is where I was working at the time, and I didn't have to worry about the four lines of text at the bottom of Page 1. Lines 330 to 470 move the data to a block of memory above Page 2. The data is stored in locations 24576 to 32255 since you only need 7680 bytes (40 by 192) to cover all of the screen. The tricky part of the program is the FOR–NEXT loops which are used to keep track of the memory location as a function of the location on the screen. There are four nested loops. I will discuss each loop, then discuss the incrementing of each loop.

The outside loop (lines 340 and 470) is for each third of the screen. The next loop (lines 350 and 450) is for sections within each third. The next loop (lines 360 and 430) is for each line within the section. The innermost loop (lines 370 and 410) is for each byte within the line. Lines 380 to 400 actually move the information from the high resolution location (PIC) to the temporary storage location (TEMP).

The increments will be discussed starting from the innermost loop and going out. The increment for each new byte is 1 (line 390). The increment for each new line is 985. This is done by line 390 plus line 420. The increment for each new section is –7079. This is done by line 390 plus line 420 and line 440. The increment of each new third of the screen is –8063. This is done by line 390 plus line 420, line 440 and line 460.

```
800  NEXT I
810  FOR SCREEN = 1 TO 3
820  FOR SECT = 1 TO 8
830  FOR LINE = 1 TO 8
840  FOR BITE = 1 TO 40
850  POKE PIC, PEEK (TEMP)
860 PIC = PIC + 1
870 TEMP = TEMP + 1
880  NEXT BITE
890 PIC = PIC + 984
900  NEXT LINE
910 PIC = PIC - 8064
920  NEXT SECT
930 PIC = PIC - 984
940  NEXT SCREEN: REM

970  REM
RING BELL AND WAIT FOR A RETURN
TO CONTINUE

980  PRINT BEL$
990  INPUT A$: GOTO 530: REM

1020  REM
SAVE THE PICTURE

1030  PRINT : PRINT "DO YOU WANT
     TO USE ";NAME$: PRINT "AS TH
     E NAME OF THE NEW FILE? (Y/N
     ) ";: GET A$: PRINT A$
1050  ON A$ = "Y" GOTO 1070: IF A
     $ = "N" THEN  PRINT : INPUT
     "WHAT NAME DO YOU WANT TO US
     E? ";NAME$: GOTO 1070
1060  GOTO 1030
1070  PRINT D$;"BSAVE";NAME$;",A1
     6384,L8192"
1080  GOTO 530
```

```
]
]LIST

10  REM

BIT CONVERTER : VAL J GOLDING

 A CALL -A.P.P.L.E.  QUICKIE

100  HOME : PRINT "ENTER 0'S OR 1
     'S FOR ALL 8 BITS. LEFT   TO
      RIGHT...": PRINT
110 MLT = 128
120  FOR PSN = 1 TO 8
130  GET BIT$(PSN): IF (BIT$(PSN)
     ) <  ) "0" AND BIT$(PSN) <  )
     "1" THEN 130
140 BIT(PSN) =  VAL (BIT$(PSN))
150  PRINT BIT$(PSN);
160 BIT = BIT + (BIT(PSN) * MLT):
     MLT = MLT / 2
170  NEXT
180  PRINT " = ";
190  PRINT BIT" $";
200  POKE 26,165: POKE 27,25: POKE
     28,76: POKE 29,218: POKE 30,
     253: POKE 25,BIT: CALL 26
300  REM

This program prints the decimal
and hex values of an eight-bit
pattern input by the user
```

Lines 530 to 550 give the options of moving the picture, saving the picture or quitting. If the quit option is chosen, the program ends. I will discuss the save option later. For now, I will continue discussing the moving of the picture. Line 630 inputs the number of lines to move up. Please note that to move down, you simply input a negative number. Line 640 shows the high resolution screen and line 650 sets PIC and TEMP to the appropriate values. Note that the number of bytes to be changed is simply 40 times the number of lines to be moved. Line 660 skips to move down (line 770) if the number of lines to be moved is negative. The new lines will be set to the background color of the picture. This color is determined by the value of the the bottom right byte (32255) for moving up (line 700) or by the value of the top left byte (24576) for moving down (line 770). The next step is to POKE this value into the memory locations just before or after the temporary storage location of the picture. This is done in lines 710 to 730 for moving the picture up and lines 780 to 800 for moving the picture down.

After these new values are in the temporary storage location, we must move them back to the high resolution memory locations. This is done by lines 810 to 940, using the same logic as lines 340 to 470. Note that the starting location of the memory to be moved is determined by line 650. Due to line 640, you will be watching the picture on high resolution Page 2 as it moves. When the move is finished, the bell will ring (lines 980 and 260) and the program waits for a carriage return to continue (line 990). After a return, the program returns to the options of move the picture, save the picture or quit.

If the save option is chosen, the program allows you to save the picture with the same name (line 1030) or a new name (line 1050). The picture is then saved under the appropriate name and you are returned to the options (line 540).

I tested this program using my original problem and it worked. I then ran a few more tests to see if it would work for a variety of conditions. It worked under all the conditions that I tried.

There are a few minor problems with the program as I wrote it. The most bothersome is the time required to move the memory. It takes about 1.5 minutes to move a picture. Of less concern is the fact that it always moves the original picture, not the last picture shown. This has a good side effect in that you can always get the original picture back by moving 0 lines.

The last problem is that when moving the picture down, you will sometimes see the background colors drawn on the previous picture as lines (once you know what this is, it isn't a problem anymore).

There are some possible modifications which would be interesting, but I did not have sufficient motivation to try. The obvious one is to rewrite the program in machine language which I assume would speed it up to an acceptable level. The next one is to use high resolution Page 1 instead of Page 2. This would solve the problem of the background lines mentioned above, but would create a problem with the four lines of text, which could be solved in a number of ways. The temporary storage could be moved higher, which would also solve the background line problem. The most interesting modification to me would be to use the same technique to move the picture sideways.

Reference.

1. Loy Spurlock, "Understanding Hi-Res Graphics," *Apple Orchard,* (Fall 1980), pp. 12-21. Reprinted from Applesauce, Vol. 1, No. 7, Oct. 1979 and *Call -A.P.P.L.E,* January, 1980.

CATALOG CORRECTION

APPLE *IIe* 80 COLUMN, EXTENDED RAM BOARD *(H80C)* **$110.00**

Quality board made especially for A.P.P.L.E. for the Apple *IIe* ONLY. The board contains 64K of memory and uses standard Apple Protocol. NOTE: $110.00 is an introductory price good only through October 1, 1983.

We Are Not PIRATES

But We Can Show You How To **UNLOCK LOCKED-UP SOFTWARE**

with our complete program listings and step-by-step instructions

YOU CAN, TOO!

core

IN-DEPTH COVERAGE of Special Interest Subjects

- GRAPHICS
- DATA BASES
- UTILITIES
- GAMES

For Hobbyists and Advanced Computists

Published Quarterly

hardcore

HARD FACTS on APPLE SOFTWARE

An Unprecedented and Controversial disclosure of closely-held

COMPUTER SECRETS

How to: Back-Up Disks
Customize
Commercial Programs
Undo Copy-Protection & More!

Published 8 times yearly

Make checks and money orders payable to.
CORE / HARDCORE Computist
P.O. Box 44549H Tacoma, WA 98444

Annual Subscription Rate: **$25.00** for 8 issues of HARDCORE COMPUTIST and 4 issues of CORE.
Canada: **$34.00** Mexico: **$37.00**
Overseas, foreign countries: **$47.00.**

Name
Address
City State
Country Zip

Payment must accompany order.
U.S. funds only. No credit cards.

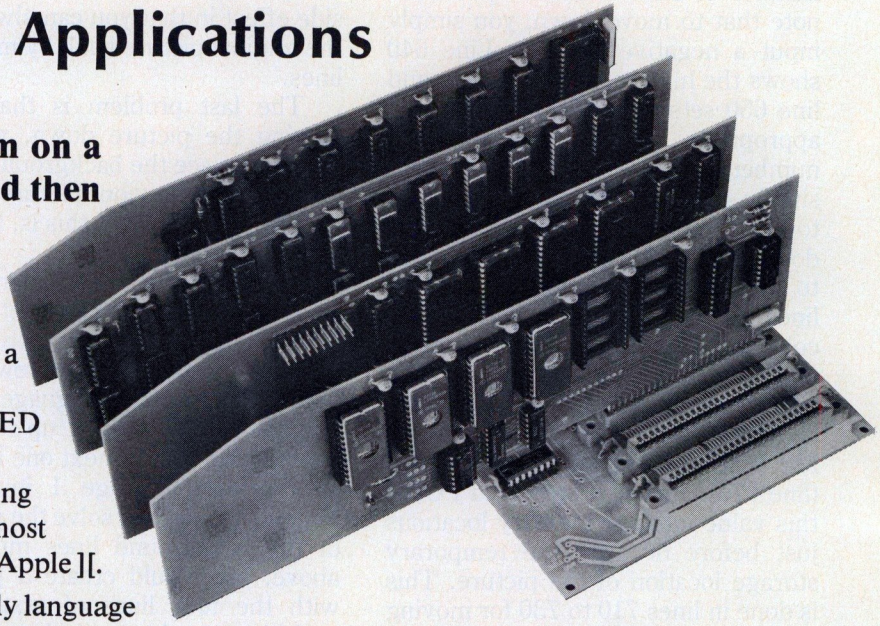
Circle 36 on Reader Service Card

APPLESEED

T.M.

Apple Compatible Hardware for Dedicated Applications

Design and program a system on a standard Apple computer and then market it with the low-cost APPLESEED system . . .



Designed for use in systems where a full Apple computer may not be necessary or desired, the APPLESEED Bus system retains all the essential features of an Apple system, including mother board compatibility with almost every plug-in board available for the Apple][.

Programs are developed in assembly language on a standard Apple][. Upon completion, a set of EPROM's are blown and inserted into the APPLESEED Bus system, which then takes control of program execution.

You buy only what you need for your application; gone are the built-in keyboard, the consumer-oriented packaging and graphics subsystems. Cost, size and power consumption are substantially reduced while flexibility is greatly increased. Compatibility with a multitude of sophisticated programming and hardware add-ons, widely available for the Apple][, has been fully retained, providing a wide range of options for designing customized systems. (Applesoft PROM's are not included.)

A sampling of APPLESEED Bus components:

CPU BOARD The 6502 has been removed from the mother board and included in this card, which also contains the clock circuitry for the system.

64K RAM BOARD 64K of dynamic RAM with provision for control to be used in the normal Apple][conventions; deleting COOO-CFFF and software control over the DOOO-FFFF area.

EPROM BOARD Jumper selectable for 2716 (EPROM) or 34199 (PROM); blow 2716 assembly language programs and then run them in this EPROM board. Jumpers allow selection of software on/off, initialize on/off and other controls.

UART BOARD Software/hardware select all standard EIA baud rates. The board supports four 6850 UART's which allow four independent input and output EIA lines. Each UART is fully programmable for parity, bit length, baud rate, etc.

UNIVERSAL BOARD The APPLESEED Universal board contains the following: One 2716 socket for an F8 PROM, one 2K × 8 bit RAM (0-7FF), one UART channel for serial communications, 8 parallel INPUT lines (TTL), 8 parallel OUTPUT lines (TTL), device decode for one additional standard slot-dependent board, support for standard Apple][style keyboard, and a power supply connection. Each feature is independent and any combination may be used simultaneously. The Universal board is designed to be used with the APPLESEED CPU board.

TEXT BOARD The TEXT board is designed to add the conventional TEXT (page 1 and page 2) to an APPLESEED system. The board generates composite video to plug into a standard monitor or T.V. modulator. The board provides 40 columns × 24 rows in each page and is not slot dependent.

MOTHER BOARD 10-slot motherboard is fully compatible with most Apple add-on cards currently available.

Please call or write for prices and more detailed information on these and other APPLESEED Bus products.

DOUGLAS ELECTRONICS 718 Marina Blvd. • San Leandro, CA 94577 • (415) 483-8770

Apple][is a registered trademark of APPLE COMPUTER INC.

Circle 17 on Reader Service Card

SHAPE TABLE SPLICER

Cyrus W. Roton

BUILDING a shape table can be simple when using one of the many utilities on the market or in the magazines, but few of those allow you to delete groups of shapes from the table or to append and insert selected shapes from a second table into the first table. With the *Shape Table Splicer* you can browse through your library of shapes and fonts, pick out just the items needed for some specific application and splice together a new table. It's easy.

Implementation

Deleting and inserting shapes in a table involves three basic tasks:

1) Sizing and locating blocks of data to be moved.
2) Relocating those blocks.
3) Incrementing or decrementing pointers.

PEEK and POKE are the natural tools for these tasks. However, all the pointers consist of 16 bit values, which require statements of the form;

```
value = PEEK(location) +
256*PEEK(location + 1)
```

Decimal values to be POKEd must be split into two bytes and POKEd into address pairs. The original version of the splicer contained many of those rather messy expressions, and it ran a bit slowly.

Thanks to Ken Kashmarek ("Taking a Bigger Byte," *Call -A.P.P.L.E.*, February 1983, p.27), a simpler form is available. The *Shape Table Splicer* makes extensive use of Ken's PEEKPOKE routine. The result is a shorter listing and a significantly faster run.

The Program

The real entry (re-entry) point is line 350, the main menu. At that point in the listing, you can see the initial options available. If you elect to merge two tables, the flow jumps to line 920, where a second shape table is to be identified and loaded. Note that the first table is loaded at address 17000, leaving High-Res page 1 clear for displaying the shapes. You will be asked to enter the address for the second table. Be sure to locate the second table well clear of the first one, which may expand as you insert or append shapes. If your Apple is short on memory, you can load shapes into High-Res page 1 and sacrifice viewing them.

There are sufficient REM statements in the listing to tell you what each section does. How they work should be easy to follow if you translate the symbols (variables) into descriptive text, as follows:

```
]LIST

10  REM

SHAPE TABLE SPLICER : CYRUS ROTON


  CALL -A.P.P.L.E. : Sep. 1983

50  PRINT  CHR$ (4)"BLOAD PEEKPOK
    E"
60  POKE 1013,76
70  CALL 768
80  GOTO 250
90  REM  * INC/DECR POINTERS
100  FOR A9 = A1 TO A2 STEP 2
110  &  PEEK A9,DN:DN = DN + J
120  &  POKE A9,DN
130  NEXT A9
140  RETURN
150  REM  * SEE SHAPES
160  &  POKE 232,SX
170 X = 100:Y = 50: HCOLOR= 3: SCALE=
    1: ROT= 0
180  HOME : VTAB 22: INPUT "ENTER
     SHAPE # (0=RETURN TO MENU)
    ";DS
190  IF (DS < 1) OR (DS > NX) THEN
     PRINT  CHR$ (7): GOTO 350
200  PRINT "X=";X;"  Y=";Y;"  O.K
     (Y/N)": GET B$
210  IF B$ = "Y" THEN 230
220  INPUT "ENTER X";X: INPUT "EN
    TER Y";Y
230  HGR : DRAW DS AT X,Y: PRINT
    "HIT KEY TO CONTINUE";: GET
    B$: PRINT
240  TEXT : GOTO 180
250  REM  * GET SHAPE TABLE FROM
    DISK *
260  HOME : PRINT "ENTER FILE NAM
    E OF SHAPE TABLE"
270  PRINT "TO BE READ FROM DISK"
    : PRINT
280  INPUT TXNAME$
290 SX = 17000
300  PRINT  CHR$ (4)"BLOAD ";TXNA
    ME$;",A";SX
310  &  PEEK 43616,LX
320 NX =  PEEK (SX)
330  HOME : PRINT "TABLE IN MEMOR
    Y. START=";SX
340  PRINT "LENGTH=";LX;" BYTES.
     SHAPES=";NX
350  REM  * MENU *
360  PRINT  TAB( 10)"SELECT OPTION"
370  PRINT "1=DELETE SOME SHAPES"

380  PRINT "2=MERGE A SECOND TABL
    E INTO FIRST TABLE"
390  PRINT "3=SEE SHAPES ON HI-RE
    S"
400  PRINT "4=SAVE TABLE ON DISK"

410  PRINT "5=LOAD NEW FIRST TABL
    E"
420  INPUT A
430  IF (A < 1) + (A > 6) <  > 0 THEN
    360
440  ON A GOTO 540,860,150,480,25
    0
450  GOTO 360
480  PRINT "WANT TO SAVE SHAPE (Y
    /N)"
490  GET B$: PRINT "": IF B$ <  >
    "Y" THEN 350
```

SHAPE TABLE SPLICER

SX = Address of start of first table (contains the number of shapes in that table).

SY = Address of start of second table.

XF,XL,XB = First and last shapes to be deleted and the number of bytes to delete.

YF,YL,YB = First and last shapes (second table) and bytes to be appended or inserted into first table.

XA,YA = Number of shapes in a block.

NX,NY = Number of shapes in first and second table.

LX,LY = Length (in bytes) of the two tables.

DS = Number of first shape to be displaced to make room for inserted shapes.

Lines 110–120 are clear examples of the simplicity afforded by Ken's PEEKPOKE routine. A 16 bit address is PEEKd, incremented or decremented and POKEd back into the address pair. It takes only three short statements.

Using the Program

Before attempting to run the *Shape Table Splicer,* you must save Ken's PEEKPOKE routine in a binary file. Line 50 loads the PEEKPOKE and line 70 initializes it. Line 60 sets the ampersand jump, which may have been changed by some previous program.

Each input is prompted by a printed question or by a menu. The "Y" and "N" (yes/no) responses do not require hitting RETURN.

Be sure to plan all your shape table modifications in advance, and remember that when you insert or delete, all the subsequent shapes are renumbered.

You can use this program just to look at shapes and to find the length of a table. Portions of the listing may be useful to you in your own programs. Be aware that if you modify and expand this program, it is likely to encroach upon HGR1. In that case, change line 230 to utilize HGR2.

The *Shape Table Splicer* has been in use for only a short time, but it has already saved me more time than it took to write it. I hope it will prove to be useful to you too.

```
500  PRINT "WHAT IS NAME OF SAVE
     FILE ?"
510  INPUT TXNAME$
520  PRINT  CHR$ (4)"BSAVE ";TXNA
     ME$;",A";SX;",L";LX
530  GOTO 350
540  REM  * DELETE A SHAPE *
550  INPUT "ENTER # OF FIRST SHAP
     E TO DELETE ";XF
560  INPUT "HOW MANY SHAPES TO DE
     LETE ";XA
570 XL = XF + XA - 1
580  IF (XF ) 0) AND (XF (  = NX)
      AND (XL (  = NX) THEN 610
590  PRINT "TABLE HAS ONLY ";NX;"
      SHAPES"
600  GOTO 350
610  PRINT : PRINT "DELETE # ";XF
     ;" THROUGH # ";XL
620  PRINT "IS THAT CORRECT (Y/N)
     ": GET B$: PRINT
630  &  PEEK (SX + 2 * XF),A6:A6 =
     SX + A6
640 A5 = SX + LX: IF XL = NX THEN
     660
650 A9 = SX + 2 * (XL + 1): &  PEEK
     A9,A5:A5 = SX + A5
660 XB = A5 - A6
670  IF XF = 1 THEN 710
680  REM  * DECR PRECEEDING POINT
     ERS
690 A1 = SX + 2:A2 = SX + 2 * (XF
     - 1)
700 J =  - 2 * XA: GOSUB 90
710  IF XL = NX THEN 750
720  REM  * DECR SUBSEQUENT POINT
     ERS
730 A1 = SX + 2 + 2 * XL:A2 = SX +
     2 * NX
740 J =  - 2 * XA - XB: GOSUB 90
750  REM  * MOVE SUBS PNTRS AND P
     REC SHAPES
760  FOR A1 = (SX + 2 * XF + 2 *
     XA) TO A6 - 1
770  POKE (A1 - 2 * XA), PEEK (A1
     )
780  NEXT A1
790  IF XL = NX THEN 840
800  REM  * MOVE SUBS SHAPES
810  FOR A1 = A5 TO (SX - 1 + LX)

820  POKE (A1 - XB - 2 * XA), PEEK
     (A1)
830  NEXT A1
840 LX = LX - XB - 2 * XA:NX = NX
     - XA
850  POKE SX,NX: GOTO 330
860  HOME : PRINT "1=NEW SECOND T
     ABLE": IF LY = 0 THEN 920
870  IF LY (  ) 0 THEN  PRINT "2=
     CURRENT SECOND TABLE"
880  INPUT A
890  IF A = 1 THEN 920
900  IF (A = 2) AND (LY (  ) 0) THEN
     1090
910  GOTO 860
920  REM  * LOAD SECOND TABLE
930  HOME : PRINT "ENTER FILE NAM
     E OF SECOND TABLE"
940  PRINT "TO BE READ FROM DISK"
     : PRINT
950  INPUT TYNAME$
960  PRINT : PRINT "END OF FIRST
     TABLE IS ";SX + LX
970  PRINT "ALLOW ROOM FOR EXPANS
     ION"
980  INPUT "ENTER DECIMAL ADDRESS
      ";SY
990  IF (SY ( 10000) OR (SY ) 370
     00) THEN 350
1000  HOME : PRINT "FILE=";TYNAME$
1010  PRINT "START ADDR=";SY
1020  PRINT : PRINT "IS THIS CORR
     ECT (Y/N)?": GET B$: PRINT "
     "
1030  IF B$ (  ) "Y" THEN 360
1040  PRINT  CHR$ (4)"BLOAD ";TYN
     AME$;",A";SY
1050  &  PEEK 43616,LY
1060 NY =  PEEK (SY)
1070  HOME : PRINT "SECOND TABLE
     IN MEMORY. START=";SY
1080  PRINT "LENGTH=";LY;" BYTES.
       SHAPES=";NY
1090  PRINT "1=APPEND SHAPES TO E
     ND OF FIRST TABLE"
1100  PRINT "2=INSERT SHAPES INTO
      FIRST TABLE"
```

```
1110  INPUT A: IF A <  > 1 AND A <
      > 2 THEN 1090
1120  IF A = 2 THEN 1530
1130  REM  * APPEND FROM SECOND T
      ABLE *
1140  PRINT "ENTER # OF FIRST SHA
      PE (SECOND TABLE)"
1150  PRINT "TO BE MOVED";: INPUT
      YF
1160  INPUT "HOW MANY SHAPES TO M
      OVE";YA
1170  HOME : PRINT "APPEND ";YF;"
       THROUGH ";YF + YA - 1
1180  PRINT "FROM SECOND TABLE. C
      ORRECT?(Y/N)": GET B$: PRINT

1190  IF B$ <  > "Y" THEN 350
1200  IF YF + YA - 1 > NY THEN 11
      40
1210  REM  * MOVE SHAPES BACK 2*Y
      A
1220  &  PEEK (SX + 2),A4:A4 = SX
      + A4
1230 A2 = SX + LX - 1
1240  FOR A3 = A2 TO A4 STEP  - 1

1250  POKE (A3 + 2 * YA), PEEK (A
      3)
1260  NEXT A3
1270  REM  * INCR PRECEEDING POIN
      TERS
1280 J = 2 * YA
1290 A1 = SX + 2:A2 = SX + 2 * NX

1300  GOSUB 90
1310  REM  * INSERT FIRST NEW POI
      NTER
1320 A1 = SX + 2 * (NX + 1):J = L
      X + 2 * YA: &  POKE A1,J
1330  IF YA = 1 THEN 1420
1340  REM  * INSERT REMAINING PNT
      RS
1350  FOR K = YF + 1 TO YF + YA -
      1
1360 A4 = SY + 2 * K:A3 = SY + 2 *
      (K - 1)
1370  &  PEEK A4,D1: &  PEEK A3,D
      2:DLTA = D1 - D2
1380 A1 = A1 + 2:J = J + DLTA
1390  &  POKE A1,J
1400  NEXT K
1410  REM  * SIZE BLOCK TO MOVE
1420 A5 = SY + LY: &  PEEK (SY +
      2 * YF),A6:A6 = SY + A6
1430 YL = YF + YA - 1: IF YL = NY
       THEN 1450
1440  &  PEEK (SY + 2 * (YL + 1))
      ,A5:A5 = SY + A5
1450 YB = A5 - A6
1460 A3 = SX + LX + 2 * YA
1470  REM  * MOVE SHAPES INTO FIR
      ST TABLE
1480  FOR K = 0 TO YB - 1
1490  POKE A3 + K, PEEK (A6 + K)
1500  NEXT K
1510 NX = NX + YA:LX = LX + YB +
      2 * YA
1520  POKE SX,NX: GOTO 330
1530  REM  ** INSERT FROM SECOND
      INTO FIRST
1540  PRINT "ENTER # OF FIRST SHA
      PE (SECOND TABLE)"
1550  PRINT "TO BE MOVED ";: INPUT
      YF
1560  INPUT "HOW MANY SHAPES TO M
      OVE ";YA
1570  IF YF + YA - 1 > NY THEN 15
      40
1580  INPUT "INSERT AFTER WHICH S
      HAPE (FIRST TABLE)";A:DS = A
       + 1: PRINT
1590  IF (A < NX) AND (A >  = 0) THEN
      1630
1600  PRINT "CAN NOT INSERT AFTER
      ";A
1610  PRINT "LAST SHAPE (FIRST TA
      BLE) IS ";NX
1620  GOTO 1580
1630  PRINT "INSERT ";YF;" (+";YA
      - 1;" MORE) FROM TABLE 2"
1640  PRINT "INTO TABLE 1 FOLLOWI
      NG # ";A: PRINT
1650  PRINT "IS THAT CORRECT (Y/N
      )": GET B$: PRINT ""
1660  IF B$ <  > "Y" THEN 350
1670 A5 = SY + LY: &  PEEK (SY +
      2 * YF),A6:A6 = SY + A6
1680 YL = YF + YA - 1: IF YL = NY
       THEN 1700
1690  &  PEEK (SY + 2 * (YL + 1))
      ,A5:A5 = SY + A5
1700 YB = A5 - A6:DLTA = YB + 2 *
      YA
1710  REM  * MOVE SUBSEQ SHAPES B
      ACK
1720  &  PEEK (SX + 2 * DS),A2:A2
       = SX + A2
1730  FOR A1 = (SX - 1 + LX) TO A
      2 STEP  - 1
1740  POKE A1 + DLTA, PEEK (A1)
1750  NEXT A1
1760  REM  * MOVE SUBS PNTRS & PR
      EC SHAPES BACK
1770  FOR A1 = (A2 - 1) TO (SX +
      2 * DS) STEP  - 1
1780  POKE (A1 + 2 * YA), PEEK (A
      1)
1790  NEXT A1
1800  REM  * INCR SUBS PNTRS
1810 J = DLTA
1820 A1 = SX + 2 * YA + 2 * DS:A2
       = SX + 2 * YA + 2 * NX
1830  GOSUB 90
1840  REM  * INCR PREC & FIRST NE
      W PNTRS
1850 J = 2 * YA
1860 A1 = SX + 2:A2 = SX + 2 * DS
      : GOSUB 90
1870  IF YA = 1 THEN 1980
1880 L = 1
1890  REM  * INSERT REMAINING PNTRS
1900  FOR K = (YF + 1) TO (YF + Y
      A - 1)
1910 A4 = SY + 2 * K:A3 = SY + 2 *
      (K - 1)
1920  &  PEEK A4,D1: &  PEEK A3,D
      2:DLTA = D1 - D2
1930 A1 = SX + 2 * (DS + L): &  PEEK
      (A1 - 2),D1:J = DLTA + D1
1940  &  POKE A1,J
1950 L = L + 1
1960  NEXT K
1970  REM  * MOVE SHAPES INTO FIR
      ST TABLE
1980  &  PEEK (SX + 2 * DS),A3:A3
       = SX + A3
1990  FOR K = 0 TO YB - 1
2000  POKE A3 + K, PEEK (A6 + K)
2010  NEXT K
2020 NX = NX + YA:LX = LX + YB +
      2 * YA
2030  POKE SX,NX: GOTO 330
3000  REM

REQUIRES & PEEK/POKE,
CALL -A.P.P.L.E. 2/83
```

The Solution to Applesoft . . .

THE ROUTINE MACHINE

Now, from the programming experts of Southwestern Data Systems, an easy-to-use way of putting the POWER and SPEED of Machine Language routines in YOUR OWN Applesoft programs!

ROUTINE MACHINE does all the work for you! No knowledge of machine language programming, whatsoever, is needed. Simply choose the function you want from an ever-growing library and Routine Machine puts just that routine in your program. Over two years in development, we've made it all completely transparent — no need to bother with BLOAD'S, HIMEM:, etc. — and it's COMPATIBLE WITH APPLE II AND APPLE IIe!

Each Library Package is so packed with useful routines that it's impossible to do more here than just list the names! So get out your magnifying glass! Call or write for more information. Better yet, order now from your local dealer. All of our software is unconditionally guaranteed!

THE ROUTINE MACHINE

The main "librarian" of the series, this package includes not only the Routine Machine itself, but all of these commonly needed routines:

Variable Swap	**Error Messages**	**Hires Characters**	**Print Using ($.00)**	**Calculated Goto**	**Turtle Graphics**
Text Output	**Calculated Gosub**	**Fast Bload**	**String Input**	**Line # Data Restore**	**Binary Address Read**
String Search	**Data Element Select**	**Reset Run**	**Array Search (1D)**	**Hex/Dec Convert**	**Reset Boot**
Bubble Sort (1D)	**Memory Move**	**Reset Onerr**	**Tones**	**Pointer Read**	
Sound Effects	**Pointer Write**	**Shape Table Converter**	**Error Handling Routine**		

$64.95*

&CHART™ — $49.95*

Library Disk

This package contains a Hi-Res chart graphics (plotting) library of routines that really helps with those custom graphics programs. Doing stock analysis, sales forecasts, scientific applications or even just plotting the monthly budget, & CHART makes the job much easier! Routines and functions include:

Page/Mode Display	**Vertical Labels**	**Zoom/Unzoom**
"Work" Page Select	**Horizontal Labels**	**FP Program Splitter**
Clipping Windows	**Grid Pattern**	**3D to 2D Transform**
Full User Scaling	**Log Scaling**	**Epson Screen Dump**
Window Reverse	**Arc Generation**	**Median Filter**
Window Clear	**Polar Charts**	**Window Frame**
Axes Generation	**Pie Charts**	**Area Fill**
Fast Hires Load	**Tic Mark Control**	**Fast Hires Save**

&ARRAY™ — $49.95*

Library Disk

This Library Package is dedicated to doing just about everything you ever thought possible (and some impossible) to an Applesoft array. With the tremendous number of routines in this package, you'll agree it's a bargain at twice the price. All routines work with two-dimensional arrays, many times with both string AND numeric data! In addition, routines are provided to perform virtually any usual mathematical operation on an entire array in ONE statement! For example, you could multiply every element in an entire array by a factor in one statement. You could also define a given array as the product of two other arrays. Over a year in development, this is a real bargain! Just some of the many routines are:

Fast Disk Read	**Array Rounding**	**USR Function Library**
Fast Disk Write	**Matrix Identity**	**Fast FP Run**
Search	**Matrix Transpose**	**Fast Brun**
Sort	**Matrix Multiply**	**Hex Poker**
Delimited Strings (1D)	**Matrix Inverse**	**Keyboard Scan**
Rename	**Array Randomize**	**FP Onerr Fix**
Clear	**Array VAL**	**RP Rnd Fix**
Redim (no data loss)	**Array STR$**	**FP Return Fix**
Row/Column Add	**Row/Column Delete**	**Fast Garbage Routine**

&SCREEN™ — $49.95*

Library Disk

An extraordinary collection of screen related routines, this disk is dedicated to text display and input for the Apple II/IIe. For starters, the Screen Generator/Screen Processor allows you to design an entire input or output screen format, and then have the entire screen executed as one or two statements. This is useful for reports, menus, data entry and more. In addition to the Screen Processor, the following routines are also included:

Input Using	**Speed Control**	**Control Char Display**
Input With Editing	**Print Pause**	**Lower Case Input**
4 Way Scrolling	**Keyboard Click**	**Special Line Scroller**
Musical Keyboard	**Printer Control**	**Center/Justify Strings**
Screen Dump	**Print Repeat**	**Clear Screen**
Memory Dump	**Memory Disassemble**	**Cursor Mouse Routine**

&SAMPLER I™ — $49.95*

Library Disk

This disk is a collection of a wide variety of useful routines written by Peter Meyer, author of the Routine Machine, and other professional programmers.

Number Input	**Catalog Modifications**	**String Execute**
If-Then-Else	**Disk File Data**	**Write Protect Sense**
Controlled Get	**Catalog to Array**	**FP Program Erase**
Print Hex	**File Selector**	**Speed Up Applesoft**
Print Binary	**FP Disk RWTS**	**Modified Chain**
String Character Strip	**Disk Sector Map**	**Slot Drv Vol Select**
String Left/Right Fill	**Find Free Sector**	**Disk Catalog Sort**
String Overlay	**Mark Sector**	**Push Button Read**
Shellsort (1D)	**Disk R/W String**	

***Plus $3.00 Shipping. (California Residents Add 6% Sales Tax.)**

More to Come!

In addition to all these Library Packages, even more are in development!

The ROUTINE MACHINE and its Library Packages really WILL make the difference in your programming! Try out these programs and if you aren't convinced that the ROUTINE MACHINE is one of the best programming innovations to come along in years, we'll gladly refund your purchase price within 30 days of purchase, with return of product.

Ask your Local Apple Dealer or Ask Him to Contact:

southwestern data systems™

10761 Woodside Avenue • Suite E • P.O. Box 582-C • Santee, California 92071 • (619) 562-3221

Circle 44 on Reader Service Card

EXTRA MEMORY FOR THE SPREADSHEET

George Mochizuki

THIS article is not for the faint hearted Apple user. It will describe the use of additional memory cards for THE Spreadsheet program. For those of you who have purchased THE Spreadsheet or the Magicalc program and need just a little more memory for your models and can't afford the monster RAM cards (64K and larger), just read on. If the reader hasn't noticed, the price of 16K RAM cards have dropped to unspeakably low prices. An example of this is Huntington Computing and their $40 RAM card; your local "hardware" store may have values just like this.

To get into the meat of this article, I have made at least one important discovery relating to the 16K RAM card. The little strap that comes with most 16K RAM cards can replace almost any 16K RAM chip. The only reason I say almost any is because I have not been brave enough to move any of the straps out of the last row of chips. When I purchased THE Spreadsheet, I also purchased a second 16K RAM card to try out.

When I finally installed the second RAM card, I decided on slot 5 for my initial trial. The first five slots were already filled with a 16K RAM card, printer, serial card, 80 column card, and a slot extender, which left slots 5 and 7 open. I then had to decide where to put the strap. The placement of the strap in its logical place was a little difficult, since the 80 column card was in the way. I had stripped my Apple of everything except both RAM cards, but I wanted to have the parts placement in a final configuration. I put the strap in the far right hand memory chip, the logical location for a RAM card in slot 7.

Following the smoke test, my Apple still worked. Turning on the cards by entering the monitor revealed that my Apple would die (a normal occurrence if there are no monitor routines in the $F800 location of the RAM card). A reset does not bring the system back to normal.

This brought me to a second discovery about the Apple 16K Language card. When the ROM space at $F800 is switched off and the RAM card is switched on without loading Integer Basic, the system will die in the exact same way. All you will get is a blank screen. I always assumed that the F8 moniter ROM on the Apple 16K RAM card was switched on even when loading Integer Basic, but such is not the case.

One extra trick I did involved splicing the old F8 ROM into the Integer Basic file from the DOS master. For those of you who are wondering about the old monitor ROM, it contains the single step and trace functions for machine language debugging. When loaded in the RAM card and turned on, these functions are available to the user.

To get back to the purpose of this article, the next set of steps involved checking the system as each card was reinstalled. I started with the disk controller card first and booted THE Spreadsheet program. With a blank Spreadsheet program, a basic 64K system will show 30K available memory. Adding the extra 16K RAM card, the program shows 46K left to use.

My experimentation worked, and I continued the insertion of the rest of the cards for my system, one at a time, checking my system with THE Spreadsheet. Everything worked harmonously. I also got real brave and purchased a third 16K RAM card. I noticed some problems associated with the Vision 80 card, three RAM cards, and THE Spreadsheet. The system seems to go off into the monitor, always in the middle of recalc (each model will have its own particular cell to stop at). This doesn't happen as often when entering data.

I talked a friend (also a brave soul) into trying all my RAM cards, Vision 80, and his Magicalc program. We observed the same exact results that occurred on my system. The problem seemed to be over driving of the address buffers, even though I had the 8T97 buss drivers in my system. These are supposed to be the real heavy duty buss drivers, but even these apparently have a fan out limit in the Apple computer.

As long as I did not use the 80 column card everything works fine, even to the extent of using five RAM cards. The "Intel" or "National" versions of the 8T97 chip (identified by the "74LS367" printed on the chip, compared to the 8T97 on the mother board next to the chip locations), do not have the same driving capability, thus you should be a lot more cautious. To be on the safe side, I now run two RAM cards on my system with the 80 column card and try to develop models that run in a standard 64K system, but it is handy to have a little extra memory available.

The recalculation time is the one real side effect to developing big models that using every bit of memory of the 16K RAM cards or the extended memory cards. It can take several minutes depending on the size of the model, and this is a good time to use the manual recalculation mode for the data entry portion of your model.

One last note, I have tested as many as five RAM cards in my system at one time (this is only for the real brave or the fool hardy; I haven't decided which yet). For this test I stripped my system: the RAM cards, and the disk controller, and it is impressive to see that much memory available for THE Spreadsheet (this test lasted only for a few seconds after booting the program).

Please note I don't recommend trying some of the things I have done. After hearing some real horror stories from dealers and other users I decided I have been very very lucky with my system. I would recommend, if you can't afford any of the extended memory cards, the addition of a second 16K RAM card. The prices are very inexpensive, perhaps $50 to $60 or less.

Hitch your Apple to a Star.

Apple, CP/M & IBM PC are registered trademarks of Apple Computer Inc., Digital Research Inc. & IBM Corp. respectively.

Now when you buy our best selling WordStar word processing software or InfoStar data base manager for your Apple II, II+ or IIe computer, we'll throw in a free CP/M board. (The one that gives you full implementation of WordStar in one board instead of three.)

With it, you'll be able to take full advantage of the advanced features that already made WordStar No. 1 on the IBM PC. And you can use InfoStar data base manager to enter, sort, and report information without programming in code.

To take us up on our offer, hurry down to your local computer store between June 1 and October 31 and pick up WordStar or InfoStar.

Either one will make your Apple shine.

MicroPro®

Technical Data: Board Includes 6 MHz Z-80B Processor, 64K RAM, CP/M 2.2, 70 Column High Resolution Capability, Shift Key Modification Device, And Expansion Interface Connector.

Circle 23 on Reader Service Card

AN APPLE FOR THE TEACHER:

GRADE BOOK REVIEWS

Kathryn Hallgrimson Suther

SCHOOL is starting, you will be kept busy teaching and evaluating students. Keeping track of individuals' progress in your class, will be an ongoing job.

In this months' column you will find two reviews of gradebook programs. Although both programs are over two years old, teachers still find them to be good products. Wes Felty's review of *The Grade Book* points out the added features this program contains. At any given time, scores can be posted using codes for privacy. Bar graphs can be generated showing grade distribution.

What about those progress reports parents like to receive? No parent appreciates a shock or surprise at report card time. This product enables you to give them a concise report. If their child is doing below average or failing, the report generated will indicate the reasons why. This is a time saving feature which we're sure educators will appreciate.

Go ahead and read these reviews. Decide upon the capabilities you want with your gradebook program. There is also information on two new gradebook programs to be found in *Tomorrows' Apples Today.*

One of the most popular programs at the Computers in Education conference at Rutgers University was *Rocky's Boots.* Those schools offering computer instruction will find this product worthwhile. It is an easy and fun way for students to discover how the computer works with Rocky as their guide.

We appreciate the educators and parents who have contributed so far to this column and look forward to your continued support. We hope to be able to publish comparative software reviews in the near future. If you can help us out in this area please let us know.

Wesley R. Felty/reviewer

The Grade Book $59.95
Condor Computing
7255 Stewart Rd
Dane, WI 53529
Not copy protected

MANY teachers are just now discovering that computers are not just toys for the math departments but potential tools for everyone. Two of the most powerful computer tools for teachers are a Text Processor and a Grade Book program! There are many Text Processors on the market, but few Grade Books. These are my findings on one of them, Condor Computing's GRADE BOOK, which I have used for two years.

When I first decided to get a Grade Book program, I felt that I would be lucky to get one working just the way I wanted by the end of the year. I am very picky about computer software and feel that it should exactly fit my needs, not that I should modify myself to fit it. I had to drive 400 miles to find the one that seemed to fit my needs, Condor Computing's The Grade Book. Imagine my surprise to find the program fully operational within two weeks. Not only did it exactly fit my needs, it changed the way I did a number of things by making jobs easy that weren't even practical to consider before.

Before using Grade Book, I wasn't able to tell students their exact grades more than once or twice a quarter, since I had to add up *every* students' scores before I could calculate any student's grade. Now I can post a list of total scores at any time (generally after each assignment is graded). The Grade Book program also generates a number of other useful reports for me, including lists of students who didn't turn in an assignment and progress reports.

The Grade Book does much more than just store students's raw scores on disk. It also calculates each student's percent grade from his scores (automatically not counting "Excused" assignments), assigns an A+, A, A- . . . F letter grade based on each class's grading standards, writes out lists of grades to put up on the wall (with student code numbers for privacy), sorts data by Name, Code number or Grade for all reports, prints out lists of students's missing work, prints out progress reports on each student, and prints out bar graphs showing the grade distributions in each period.

Grade Book Features

☐ User doesn't need to know how to use a computer.
☐ Always a control menu on screen.
☐ Good error trapping.
☐ Stores students's raw scores.
☐ Easy editing of scores and grades.
☐ Uses a student code number to maintain privacy in posted grade printouts.
☐ A set of test data is included to help learn to use The Grade Book.
☐ More than one class can be stored on the same disk.
☐ Allows data to be sorted in terms of Names, Code numbers, or Grades.
☐ Allows "Excused" assignments (they are not counted in grading a student).
☐ "Flags" student's name doing less than average or failing work.
☐ Prints out a list of students who did not turn in any given assignment.
☐ Calculates current assignment grades, quarter grades, and semester grades.
☐ Assigns letter grades A+, A, A- . . . F automatically.
☐ Allows a change in the letter grade cutoff percents at any time.
☐ Prints out progress reports on any student or all students to the screen or printer. Reports include a list of all assignments, quarter grades, and comments.
☐ Progress report comments can be picked from a list of 20. These may be changed for each class. Teacher can also add own comments.
☐ Grades may be modified several ways. After grades are calculated, they can still be modified and recalculated.
☐ Weighing factors can be used to change the relative "weights" of each assignment.

GRADE BOOK REVIEWS

To calculate semester grades, you pick what percent of the grade you wish to be based on 1st quarter, 2nd quarter, and semester exam. You may change these values after scanning the grades.

Most grade book programs on the market don't give you half of these options or the flexibility to go back and make changes after the grades are calculated.

How The Grade Book Is Used

The Grade Book program is a data base program that uses 14 programs, such as ENTER STUDENT'S NAMES, ENTER SCORES, CALCULATE GRADES, etc, each do different things to the stored classroom data. The user doesn't need to know this since control menus do all of the switching from program to program, but this knowledge makes it easier to understand what is going on at all times.

First of all, the user enters all of the students's names. A code number is also entered so later classroom printouts can maintain students's privacy. The Grade Book will alphabetize these for you if you wish. These names are saved to disk, but can be later edited with changes, additions, or deletions. After the names are entered, the Grade Book will print out class lists for use for those first weeks of musical chairs. As students are added, moved, or dropped this feature is handy to reorganize and print out new class lists with attendance blanks.

The most used program is "ENTER SCORES." Each time new grades are entered into the teacher's grade book, they can be added to the computer. I find that it takes about five minutes per class to enter new raw scores. The job is made easier by the fact that the students are listed in the computer in exactly the same order as in my grade book, even when late arrivers in the class are not in alphabetical order. Along with the raw data is entered the maximum points possible and a weighing factor. These numbers allow the Grade Book to calculate percent grades and give more .or less emphasis to each assignment. A special code is entered for missing or excused assignments so that the computer can print out lists of these. It is very easy here also to later go back and change scores if errors are found or missing work is turned in.

The next most frequently used program is CALCULATE GRADES. This program does just what it says and then lets you scan the students's grades and go back and modify the grading curve two different ways. You can also print out a list of grades that include the last assignment entered and the total grade at this time for each student. This listing can be by students's names or code number and can be unsorted or sorted by Name, Code number, or Grade. Whenever I calculate grades, I print out an unsorted copy with names for my grade book and a sort by grades copy with code numbers for putting up in the classroom. Students always know the day after a test what they received on the test and what their total grade is. This eliminates surprises at the end of the quarter.

One of the Grade Book's most valuable features is its PROGRESS REPORTS. This program allows the user to see on the screen or print out on the printer any number of copies of a student's progress. It lists all of the student's grades on assignments, compares them to the class average (if the teacher wants to), and prints out the student's total grade for each quarter. It also allows comments to be printed from a list, of 20 (the teacher can use the supplied list or change them).

This feature has been of tremendous value to me. If a student, counselor, or parent asks how a student is doing, these progress reports have a lot of impact. To say that a student is failing doesn't really say much, but to show a list of all of his/her grades and show that he only turned in 4 out of twenty assignments shows exactly why. I have received a lot of favorable comments from parents and counselors about these reports. In fact, the only people who sometimes don't like them are some of the students. They quickly destroy most alibis.

The last program on the disk, CALCULATE SEMESTER GRADES, allows you to determine what percent of the total semester grade should be based on each quarter. The grades are then calculated and displayed on the screen. If they are not acceptable, they can be recalculated over and over again. The "curving factors" can be individually changed for each of the four grades (two quarter grades, semester exam grade, and total semester grade). When the grades are acceptable to the user, they are printed out in exactly the same order as the students in the teacher's actual grade book.

In all of the programs where data is stored, there is a built in way to save the data also to a backup disk. After asking if you wish to back up your data, the Grade Book uses a series of screen prompts to lead you through the process. If you find that your class has filled your disk with peanut butter (I've seen it happen), or you drop your disk on the floor and roll over it with your roll around chair as you reach down to pick it up, (I've seen that, too) you can still easily have all of your data intact on a backup disk.

There are several other programs in the Grade Book, but I won't go into them here since they are used only once usually to meet the user's individual standards. One of the greatest strengths of the Grade Book program is its flexibility. You can always come back and change any data entered, including students's raw scores, letter grade cut-offs, and weighing factors. After entering data, you can calculate grades, scan the grades, and then go back and change the way the grades are "curved" until you are satisfied. All of these changes would not be practical without the high speed aid of a powerful computer program like Condor Computing's Grade Book.

Ease of Use

The Grade Book is fairly easy to use *after* you understand it. It does many things and does them very well with options (such as sorting by names, code numbers, or grades).

Therefore, it is somewhat complex to use at first. But with more options and more "user friendlines," room to store data is lost. Therefore a compromise must be made. The Grade Book has error trapping to prevent the program from "bombing," good control menus, and good documentation; but doesn't go to great pains to help the user from inside the program. A set of practice data is included on the disk, but no tutorial. The documentation has a table of contents and is well organized to lead the user into and through using Grade Book. The program and documentation assumes that the user is *not* a programmer and may not be too familiar with a computer. The documentation should be read (unfortunately, teachers many times neglect reading directions).

Limitations

The "standard" version of The Grade Book has room for 130 students and 20 grades per quarter, for two quarters, therefore handling 40 grades per semester. This reaches the limits of a 48K micro computer's memory. Two new versions are being tested presently. One of them will allow more grades if a teacher has fewer students (like 70 students and 40 grades per quarter). The other version will allow 165 students and 30 grades if used with a RAM card or Language card on an Apple][or with an Apple *//e*.

Condor Computing sells the Grade Book for $59.95. A school can get a licensing agreement for unlimited copies for $250 (just four teachers going in together could equal that). The Grade Book comes with documentation in a three ring notebook and two disks. The disks are not at this time copy protected, so the user can make more backup copies. The author, John Faludi, is a teacher himself, so he really knows what a teacher wants and needs. He is willing to listen to user input. Several embellishments came from customers (part of the reason for leaving it unlocked is so that you can play with it, but it is a *very* complex program). The program, like students, continues to grow. When users needed more assignment capacity, the Grade Book grew into it. When the Apple *//e* came out, Grade Book moved in. As needs and machines change, so does Condor Computing's Grade Book.

Pat Buck/reviewer

Gradebook Plus
Individual Copy *$49.00*
Schoolwide Copy *$125.00*
Educational Microcomputer Associates, Inc.
EMA
Box 339
Los Altos, CA 94022
(415) 326–1585
Copy protected
User can make multiple copies

EVERY teacher needs a secretary! Most teachers will agree with that statement since just taking charge of 25 or more students, presenting lessons, maintaining control, evaluating progress, keeping up with the curriculum, counseling disturbed or frightened children and parents, etc. are just a few things a teacher will do. We are also expected to teach manners, courtesy, social values, brotherhood, respect for people and property, and so forth. Most of us who have been around the track a few years do all of this automatically, but the paper work gets us down! Ideally, a loyal secretary or a classroom aide would be the answer, but that is just not possible for most of us. Any kind of help with record keeping is a necessity. I have found the answer to my dream in an excellent grade keeping software program called *Gradebook Plus.*

I first ran across this program at the CUE (Computer Using Educators) Conference in the Fall of 1981. In fact, this is the first software program I ever bought. I have not seen this advertised in any magazine nor have I seen it displayed in any computer software store. I think it may be the best kept secret in the computer world. As far as I know at the moment it is only available from the software publishers as listed above.

The first thing a teacher must do in using any kind of grading system with hard statistics to back him up is to decide just how he wants to grade his students. In my own case, teaching math and computers in a junior high school, it was relatively easy. Math teachers have an easy time of grading anyway — the numbers tell it all. I decided that in each of the four grading periods in the school year, major or chapter tests would count 50%, pop quizzes would count 25%, and the remaining 25% would be in evaluating daily work, homework, attitude and effort. The test, of course, is completely subjective, but the rest is not. With 1000 points, give or take 50–100, each quarter, I could count on 5 tests each worth 100 points, 5 pop quizzes each worth 50 points, and 5 notebook checks each awarded 50 points.

The Gradebook Plus fits into my scheme just perfectly.

The Gradebook Plus has an ingenious copy protection plan that allows the buyer to copy as many times as needed, but eliminates the possibility for others by using a password protection scheme. This prevents the buyer from selling or giving copies to others because the first user's name is indelibly printed on every subsequent copy. I use a copy disk for each quarter and save the disk for a full year before I overwrite it in case some unforseen controversy about a grade arises later.

There is a schoolwide version of the Gradebook Plus which allows every member of the faculty to use the program.

Each disk can hold six classes, 40 students per class, 60 entries per class, 999 pts. as the largest score per assignment, and 9999 points as the total points per student. Obviously this is ample room for even the most dedicated teacher!

The documentation is superb! The authors have taken it into account that many teachers are not yet so computer literate that they can approach a computer with the same ease and familiarity as they face their captive audiences each day. Yet one who has computer experience can skim the documentation in a brief 15 minutes and learn all he needs to begin using the program. There are several appendices with models of every printout that can be made. Program instructions are also on the disk to be referred to when needed.

The utilities component allows the user to choose the options he/she needs to get started. There is a choice on which model Apple one is using, what kind of printer, what size printer paper, whether the printout is to be regular size or compressed print, the usual printer slot, etc.

GRADE BOOK REVIEWS

The main Program Menu has the following options:

1) INITIALIZE CLASS FILE
2) ENTER CLASS SET OF SCORES
3) CORRECT, ADD, OR DELETE ENTRIES
4) DISPLAY SCORES OF A SINGLE STUDENT
5) DISPLAY GRADEBOOK OF ENTIRE GLASS
6) DISPLAY STATS OF SELECTED ENTRIES
7) BACKUP AND UTILITIES
8) PROGRAM INSTRUCTIONS
9) EXIT PROGRAM

This program is menu driven, user friendly, with appropriate instructions along the way.

After scores have been entered, at any time a teacher can produce a current dated printout of a single student or an entire class with a grade assigned to the score which is averaged out. The teacher tells the program just exactly what grades to assign to what scores or averages. He/she may do this once for this class, and the computer will assign these grades always, or he/she may change them each time. Assignments can be weighted differently, and extra credit assignments are easily taken care of. If a student misses an assignment or test and somehow or other doesn't make it up (if illness is a factor, for example) the teacher can either count this as a zero or average out the other scores as if this one doesn't exist. This is a fine option depending on the circumstances of the missed assignment.

This program has been like an efficient secretary for me. I use option #3 in the Program Menu "DISPLAY SCORES OF A SINGLE STUDENT" when I want a dated printout of a student's scores up to the minute when I have a conference with his/her parents. There is no doubt in the parents' minds just exactly how their child is doing — or in the student's! Option #5 "DISPLAY GRADEBOOK OF THE ENTIRE CLASS" is invaluable when I must record grades on report cards at the end of the quarter. I also use the "SINGLE STUDENT OPTION" to give a print out to every single one of my students three weeks before the grading period comes to a close so that he/she knows exactly where he/she stands and can act accordingly. I have very competitive, turned on students, but there is never an argument about a grade. A print out seems to hold the "aura of truth" for the computer wise modern child! It is, therefore, official.

Although I have used this program with the easiest application, (mathematics) the least subjective of the disciplines to grade, I am confident a teacher in other subjects or with self-contained classes in elementary schools could use it as easily as departmentalized high schools or junior highs. As a person who is used to grading exams and homework, I would have to give an A to this entire package. It is a job well done, easy to use, and witch a flexible approach can be adapted to any teacher's subject or style of grading.

FOR YOUR APPLE**

80/132 COLUMN CARD $99*/$129*PPD

Expand your APPLE with 132 COLUMN display (20MHZ Monitor - Need for 132 MODE only)
Card comes standard for 80 COLUMN display with an optional 132 COLUMN capacity.
True desenders and compatible with VISICALC** with a preboot diskette. If card ordered with the 132 COLUMN option $129.00.
Basic 80 COLUMN $99.00. 132 COLUMN option can be ordered later for $29.50. Preboot for VISICALC** add $24.50. Shipping date 9/15 /83.

II**e 64K 80 COL CARD $99*PPD

Expand your IIe** with 80 COLUMNS and 64K MEMORY

e-Z CARD $60*PPD

Uses Z80A CPU (Needed for WORDSTAR**, D BASE II**, MULTIPLAN** and runs all 2.2 CPM** PROGRAMS
Software not provided.

128K CARD w/Built-in Disk $209*PPD

The ORBITAL 128K RAM CARD with a disk drive built-in, will expand the memory of your Apple* II, II+, IIe another 128K or can be used as a second disk drive.
The ORBITAL 128K CARD emulates a disk drive but, not like the standard mechanical drives the ORBITAL 128K CARD is ten times faster.
It is faster because it fits inside the Apple* just like any other card and has no motor to turn on or off.

*ADD $30.00 IF YOU WANT A READY TO USE UNIT

ALL cards come in KIT form and have ALL REQUIRED IC's, Printed Circuit Board, and Assembly Instructions.

GUARANTEE IF YOU ARE UNABLE TO GET YOUR KIT TO WORK, for a fee of $30.00 plus $2.50 postage and handling (for socketed boards only, not provided), we will correct any errors in soldering and or blown chips, and mail your card back ready to use.

Orbital Systems, Inc

ORBITAL SYSTEMS INC.
2929 E. JASMINE ST. MESA, ARIZONA 85203
1-800-227-9275 EXT. 950- A 24 HR TOLL FREE 7 DAYS A WEEK
ARIZONA RESIDENTS CALL COLLECT FOR ORDERS 602-830-6457 9 A.M. TO 3 P.M. MONDAY THRU FRIDAY

Postage paid does not apply to international or C.O.D. orders. Shipping in USA through UPS.

MASTER CHARGE AND VISA ACCEPTED
Arizona residents add 6% for sales tax

Dealers Inquiries Invited

**CPM trademark Digital Research
**Multiplan trademark of Microsoft
**D Base II trademark of Ashton Tate
**Apple is the trademark of Apple Computer Inc
**Wordstar is a trademark of Micro Pro Inc
**Visicalc is a trademark of Visicalc Corp

Circle 32 on Reader Service Card

A.P.P.L.E. COMPUTER CAMP '83

Charlie/Susan Stillman

LAST summer A.P.P.L.E. sponsored a computer camp for junior high and high school students at Fort Worden in Port Townsend, Washington. Staffed by six club members, the camp was run under the aegis of Centrum Foundation. Centrum provided the facilities: housing, food, classrooms, dorm counsellors, recreation, etc. We were responsible only for most equipment and all the curriculum. All the organizers viewed this first attempt as a trial run. Nobody knew exactly what to expect. We had 18 Apple][’s and 64 kids, a core curriculum in Applesoft programming and applications projects. From the first day, students kept the two chairs in front of each Apple hot whenever the computer room was open (9 AM to 9:30 PM). Students kept us hopping nightly designing curriculum to cover the incredible spread in abilities. When it was over, the primary complaint on the campers evaluations was *not enough computer time!*

This year we returned for two weeks of a similar workout. A week for beginners was followed by a week for advanced programmers. By raising the proportion of computers to students to one for every two students, and keeping the computers on from 9 AM to 10 PM every day, we found that some kids even left the computer room voluntarily before we kicked them out at closing. Others actually went swimming on their own initiative.

Currently operated as a state park, Fort Worden is a large and fascinating place to explore: rows of gun batteries, a marine science center on the dock, and extensive beaches and trails. (The film “An Officer and a Gentleman” was made here.) We threatened to close the computer room on sunny afternoons for a couple of hours, but even then we were ambivalent about cutting kids off from the computers. This was after all a *computer* camp[1], and there was limited interest in the fort’s other facilities. Many of the participants have scant time with computers away from camp. They may get only a few minutes a week on the keyboard at school, and many do not have a computer at home. Yet the best teacher of computing is the computer itself. It lets the student make the same mistake over and over without losing patience, is very consistent, and never gets tired, which is not to suggest that supervision is unnecessary. To encourage programming, we had to ban game playing before 3:30, and we needed to suggest programming tasks for some to get them started.

What did we feel the campers should know and what did they want to learn? A desire to program games dominated the beginner’s week. Most recognized early on that writing the next Space Invaders was beyond them for now, and they launched into writing adventure or strategy games where speedy graphics were not needed. Some of the most successful programs started with a simple idea and were refined throughout the week. One favorite started as a simple Low-Res box, then a house. Later a tree and some clouds were added. Next, one cloud was animated...and ZZZAPPP!!...a lightning bolt struck the tree! Finally, came sound to enhance these special effects.

We felt a responsibility to teach BASIC and some of the computer’s practical applications. Everyone was introduced to the easy-to-learn *Zardax* word processor program. Several kids produced amusing form letter paragraphs for a “letter home.” One girl interested in becoming an astronaut used Zardax to refine a letter which she later sent to NASA. A copy of the venerable File Cabinet database program was given to everyone. At the end of the camp, they each could take with them a name, address, and interests file created by all the campers. In addition to these fundamentals, we providing the kids with a variety of tools to make their programs aesthetic successes. For example, we offered a variety of approaches to graphics. One approach was through Logo’s Turtlegraphics, an easy way to produce pleasing geometric patterns. Another side of Logo was also demonstrated via a unit on random poetry. Yes Virginia, Logo *can* be used for word manipulation too! The kids quickly adopted the tools we offered. For example, they were soon combining static images created by a graphics tablet with animation effects done in High-Res graphics using the shape table editor. For those willing to experiment, Dave Lingwood’s “AMPTONE” utility (*Call -A.P.P.L.E. in Depth 1:* All About Applesoft) gave them the sounds they wanted.

Instead of enforcing play time outside the computer room, we integrated the fort into three computer activities. Early in the week, a scavenger hunt turned sober, sedentary victims of the green screen’s magic attraction into normal sweating panting thunderous teenagers. They raced back and forth from the depths of the mule barn with the patent number on a toilet or to the gun batteries looking for a particular code. Clues were designed to relate to the Apple so that knowledge of it could save steps.

The beginners were required to do more running than thinking, but our devious scavenger hunt designers outdid themselves for the advanced group. Those teams were sometimes stumped on a problem for a half hour at a stretch. One particularly nasty Boolean algebraic expression required the Print Shop building number for solution....or so it seemed. Actually, careful examination of the expression revealed that the building number was irrelevant! Each team gets clues in a different randomized order. A team’s play continues until (1) all clues are answered or (2) someone tries to exit the program.

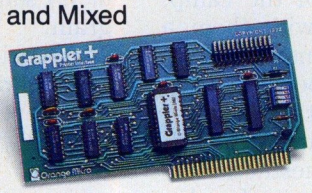
We tried to convince the kids that the program was written in a hyper-high-level language which automatically produced the clues for us!

[1] (The American Camping Association, by the way, would be quick to challenge our use of the word “camp,” but somehow the title: “Intensive Live-in Computer Workshop” doesn’t ring true either. After all, it wasn’t work for the kids.)

Satisfying Over With Innovation

Grappler™ + Printer Interface

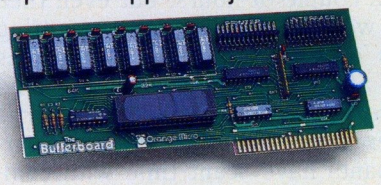
The Original Apple® graphics printer interface. Since its introduction three years ago, the Grappler has been imitated by many, but never matched. Now with Dual Hi-Res Graphics for side by side reproductions and Mixed Mode screen dumping, the Grappler + remains the most intelligent interface available. Over two dozen commands give Apple users full control over any graphics or text on the Apple screen, including a new 80 column text dump. Performance, reliability and support have made the Grappler + the #1 selling intelligent Apple interface.



Bufferboard™

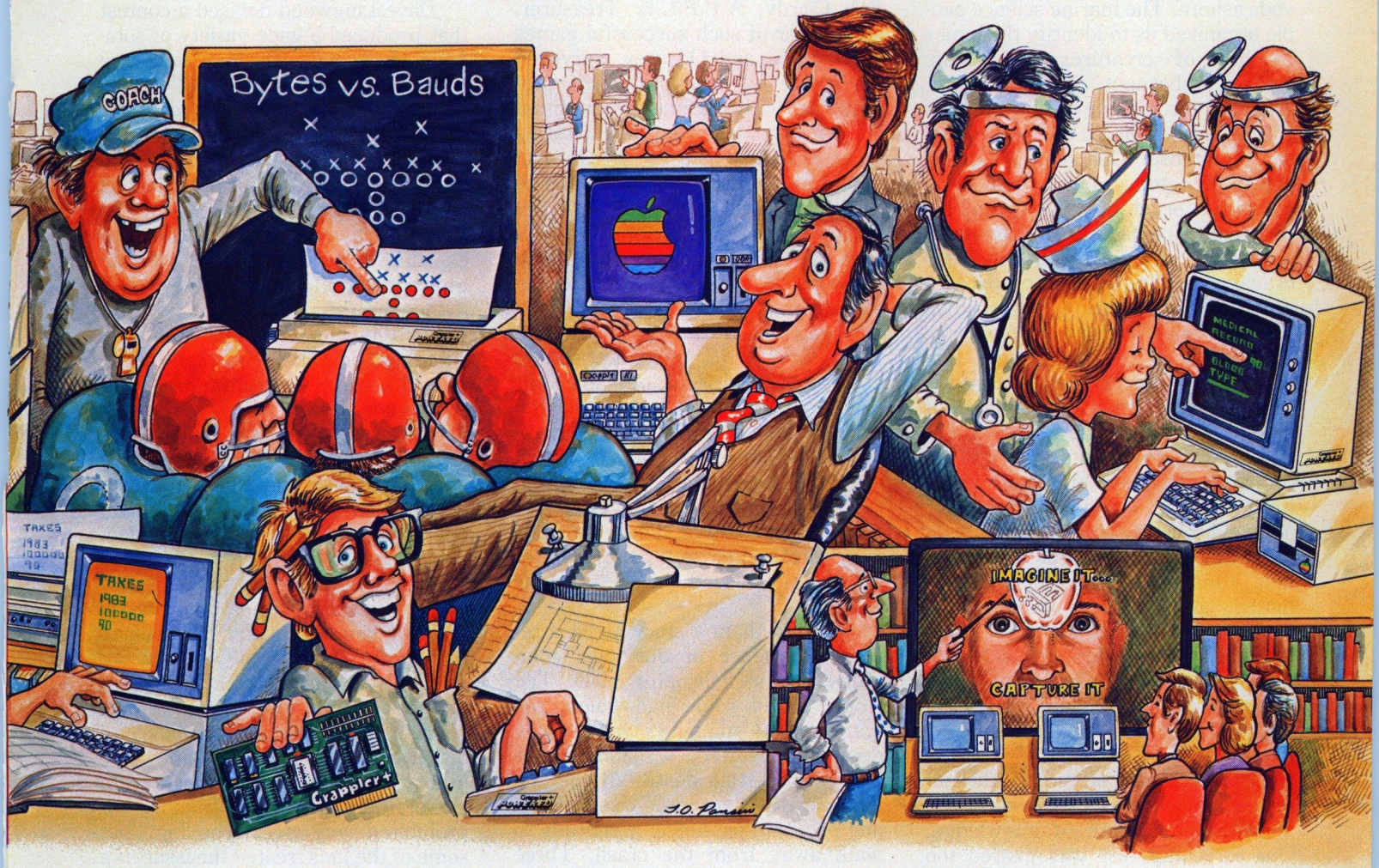
For Apples and Printers

The economical way to add printer buffering. The Bufferboard easily adds memory to your current Apple interface system, freeing your computer for additional input. Easily upgradable from 16K, the Bufferboard can store up to 20 pages of text. It fits neatly inside your Apple, "docking" onto your existing printer interface.* No clumsy boxes or cables, no external power supplies…just convenience and economy. With the Bufferboard, you might never wait for your printer again.



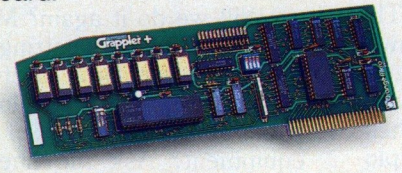
*Versions for standard Grappler + , Epson APL and Apple Parallel Interfaces.

90,000 Apple® Owners And Excellence.



BUFFERED Grappler + ™

The most sophisticated buffered Apple printer interface available. The New Buffered Grappler + combines the industry leading features of the Grappler + with the time saving economies of the Bufferboard.**



With this much interface power, you'll never need anything else. And the price will surprise you, too.

**Not available for IDS printers.
Apple is a registered trademark of Apple Computers, Inc.

Over 90,000 Apple computers are using Orange Micro products. Innovation and excellence have made us the #1 manufacturer of intelligent printer interfaces. The top selling Grappler + has become an industry standard, recommended by more software houses and Apple dealers. To meet the users' latest needs, Orange Micro will continue to introduce new products. Recent innovations include the Grappler + for IDS color printers and the new Orange Interface, with text screen dumps and formatting at a low price. There is an Orange Micro product designed for your application.

For a complete demonstration, see your Apple dealer today.

Orange Micro inc.

1400 N. Lakeview Ave., Anaheim, CA 92807 U.S.A.
(714) 779-2772 TELEX: 183511 CSMA

©Orange Micro, Inc., 1983

Circle 28 on Reader Service Card

COMPUTER CAMP

Two projects brought campers into contact with the marine center and the beach. The first, beach seining, is a method of surveying the marine life in the area immediately off the beach. A boat positions a long net in a wide arc off the beach, and teams drag the ends ashore. The marine science people organized us to identify the major families of creatures we found: sculpin, tubesnout, shrimp, perch, flounder, etc. One never knows what will be brought in, so the excitement can be intense. This year, we didn't pull in an octopus, but several of us will never forget the 1,600 shrimp we counted! Once back to the computer room, we used the graphics pad for drawings of the fish, and prepared a report of frequencies by type of fish for the marine sciences people.

The second beach project was dubbed the "critter count." A grid is laid out on the beach and a number of sampling regions are randomly chosen from the total area. Each kid drops a rounded-out wire coat hanger as a delimiter for his/her assigned observation area, while a surveying transit records the elevation. Amount of vegetation is estimated, and number of other life forms and surface features recorded by type. Once the data are collected, they are entered into the Apple statistics package. Then an overall analysis can be run and specific hypotheses can be tested.

The students attending the intermediate-advanced week surprised us with their depth of knowledge. Our screening test was geared too low, but we found that many of them knew enough about the Apple to tell us what they needed to learn anyway. This caused some last-minute curriculum planning on the part of the staff. About half the students wanted to learn machine language, and many of the rest wished to learn advanced BASIC techniques. We were fortunate to have on our staff people with sufficient depth to meet their needs. Few had "serious" applications in mind, although one individual stopped by daily to update his CP/M database after telephoning his stock broker.

As we expected, the campers were especially intrigued by graphics. The two inexpensive Koala pads ($125 retail) were in constant demand, and the art produced was very creative. The shape table editor was also heavily used. One advanced student devoted much of his week to refining this program for everyone's benefit.

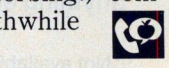
We drew on some other local talent to show the kids advanced graphics techniques and applications. Bob Clardy, A.P.P.L.E. Treasurer, and author of such successful games as Microbe and Odyssey, surveyed the range of graphics techniques available to the game programmer and left us with a diskette illustrating those techniques all free for the asking. Another evening, Bill Miller of Shelter Research Institute showed us a more serious side of Apple graphics with his Imaginator 3-D graphics package he has developed for designers and architects.

Assignments to try out particular facets of BASIC were not always assiduously carried out by our campers, but this year we ran several contests which really sparked motivation. We were teaching the use of Applesoft shape tables, renumbering, "&" calls, and the principles of program organization. When the kids were encouraged to try out some of these functions, the response was lukewarm.

We told them a story about a recent experience of our local commuter traffic helicopter pilot. The 'copter's tail rotor had failed; the craft started spinning. On the way down, it inverted, threatening the pilot with certain death. At the last moment, the helicopter was righted, and pilot and passenger were able to walk away from the crash. Three hours later, the situation finally caught up with the pilot and he collapsed. A week later, he was back on the job. We offered a box of diskettes for the best reenactment of that event. The best entry was put together by a team of three kids. One created the backdrop on the Koala pad (Seattle Space needle in the background), another developed the shape table for animation, and another worked on the sound. The helicopter falls, turns upside down, rights itself, hits the ground, a figure runs out, falls, and ambulance with klaxon takes him away. The helicopter is engulfed in a fiery explosion...pause...switch to text screen . . . "Helicopter crashes. Film at 11PM." Judging was to be on cleverness and attention to detail as well as program structure. The use of REM statements and subroutines were required. There was compliance...just barely...we found a GOSUB to a routine that had some REM statements and a RETURN at the end! Oh well...so much for program structure!

Dave Lingwood devised a contest that produced a wide variety of solutions. He wrote a very inefficient program called "SLOWPOKE" which took 40 seconds to perform computations on an array of one hundred numbers. This task was designed to test their knowledge of optimization strategies just presented in class. The winning rewrite took .8 seconds to execute. In order to obtain accurate time measurements, it was necessary to increase the size of the numeric array to 1000 elements. One individual submitted an entry which did that in only .1 seconds . . . hmmmm. A look at the code showed why. There was a mysterious ampersand call at the beginning and the end. What did they do? They caused program memory to be modified directly so that only *100* operations were actually performed. At the end, the number was restored to 1000. Another "slowpoke" entry not completed was a rewrite in machine language with direct calls to Applesoft floating point routines.

The only sobering experience of the entire camp, was the piracy problem. Kids showed up with boxes of diskettes with pirated game programs which they were prepared to share freely with their friends. We prohibited any such copying, but some of the kids treated the issue as a game. We fought back the urge to thumbtack to the wall any such disks we found (several of us have written commercial programs). We talked with the campers about this at length during the second week when the problem was most severe. We applauded those who stayed away from piracy, and lamented the effects of piracy on the industry and on the authors who are being hurt.

Will we do it again? Maybe. The biggest problem is getting the equipment together. How carefully did we drive our aging VW bus loaded down with $20,000 worth of computer equipment? Ask the line of traffic that was behind us! Being around kids absorbed in (absorbing!) computing made it all worthwhile 

A Call —A.P.P.L.E. SOFTWARE REVIEW:

ROCKY'S BOOTS

Earle Hancock/reviewer

Rocky's Boots *$45.00*
The Learning Company
545 Middlefield Road
Suite 170
Menlo Park, CA 94025
Copy protected
One backup only: *$15.00*

MY June issue of *Call –A.P.P.L.E.* came today. As I sat reading it, I was surprised and delighted to come to a section devoted to education. Apples have been a major part of my special education classroom for more than three years. Now *Call –A.P.P.L.E.* can be even more helpful to me, in a different way, as an educational resource. It's no secret that good educational software is hard to find. So when I discover an excellent program, I like to share the news with others. Rocky's Boots is my latest favorite.

Learning how the computer works, deep in the dark reaches of the processor, is easy and fun with Rocky as your guide. Rocky's Boots is a delightful simulation that introduces the elementary logic necessary to computers. Logic gates AND, OR and NOT, are presented along with beginning electronics by a whimsical character named Rocky. The learner is required to design logic circuits that solve problems presented by Rocky. After learning the logic gates, you learn to build more complex "machines" using CLOCKS, FLIP-FLOPS and DELAYS. Finally, you are able to design your own games using Rocky's game editor.

The initial screen is a demonstration which you can exit by pressing the space bar. At this point, the sound may be turned off by [Ctrl–G]; soundless programs are much appreciated in a room full of computers. The menu is next.

Please choose:
1) How to Move
2) Building machines
3) Logic Gates
4) Rocky's Boots
5) Flipflops
6) Rocky's Challenge
7) End

(Rocky suggests: play them in order.)

Choose the one you want and type its number. Numbers 1,2,3, and 5 are sections where you learn what is needed in order to "play." Numbers 4 and 6 are sections where you can build very complicated machines.

Each of the sections build on ideas learned in previous sections. It is, therefore, advisable to work the sections in order. Once mastery is achieved in a section, you need not spend your time on it again.

Moving through the lessons and building the machines is facilitated by use of the I diamond. This is the same as J✣K the cursor control commands M in Applesoft and is used by many software packages. Pressing the I moves the cursor (in Rocky's Boots the cursor is a large orange, electrically charged square) one space upward; pressing the J moves one space to the left, the K to the right and the M down. A joystick can be used in place of the keys. Pressing the space bar picks up the object under the cursor and pressing the space bar again drops same. At any time during the lesson, the ESC key gets you back to the menu. These commands make sense and are easily remembered.

Rocky's Boots is easy to use, educationally sound, and as far as I can tell, bug free. Students, from seven to adult, seem to enjoy Rocky's Boots and they learn from it.

No software is without drawbacks of some kind and Rocky's Boots is no exception. The diskette is copy protected, therefore, back-up copies cannot be made. One back-up is available at a cost of $15 with a coupon enclosed with the original. In my opinion the purchase price of $45 should include a back-up. Further, the protection scheme occasionally causes problems booting the diskette. These drawbacks are shared by many programs and are symptomatic of industry paranoia about software protection. Even with the problems mentioned, this program is highly recommended.

...about the author

Earle Hancock has directed the microcomputer project at Minuteman Regional Vocational School for three years. Prior to developing this position, he worked for Minuteman Tech as a math and special needs teacher. His 14 years in education have mostly been spent working with students who suffer from moderate to severe motivational problems. Mr. Hancock is a graduate of Boston University School of Business and holds an advanced degree from Boston College School of Education. He has served as an advisor to the Massachusetts Association of Vocational Administrators and is an active member of the Boston Computer Society and other computer organizations, including Call –A.P.P.L.E.

EDITOR BYTES BACK

Continued from page 5

Speaking of telecommunications, **Bruce Bourbonnais** describes some of his own experiences in starting a business in **Electronic Mail: Another Use for the Apple.**

Dan Lyons, after many, many tries finally saw print in our pages, and we know you'll find it intriguing to read **How I Became a Call -A.P.P.L.E. Author.**

And my, oh my, oh my, we've really overextended ourselves again (our editor may cut us off), so look for yourself on the contents page and see what else we've packed this issue with

Peace.

How to quickly re

If you'd like to turn the agony of small business bookkeeping into the ecstasy of total control, you've come to the right place.

Because even if you're starting with a shoe box full of invoices or a pile of checks hiding under a pile of deposit slips, we can tell you how to centralize, organize and monitor all that information, and manipulate it in ways that will make your business a pleasure—all with an Apple® III Personal Computer.

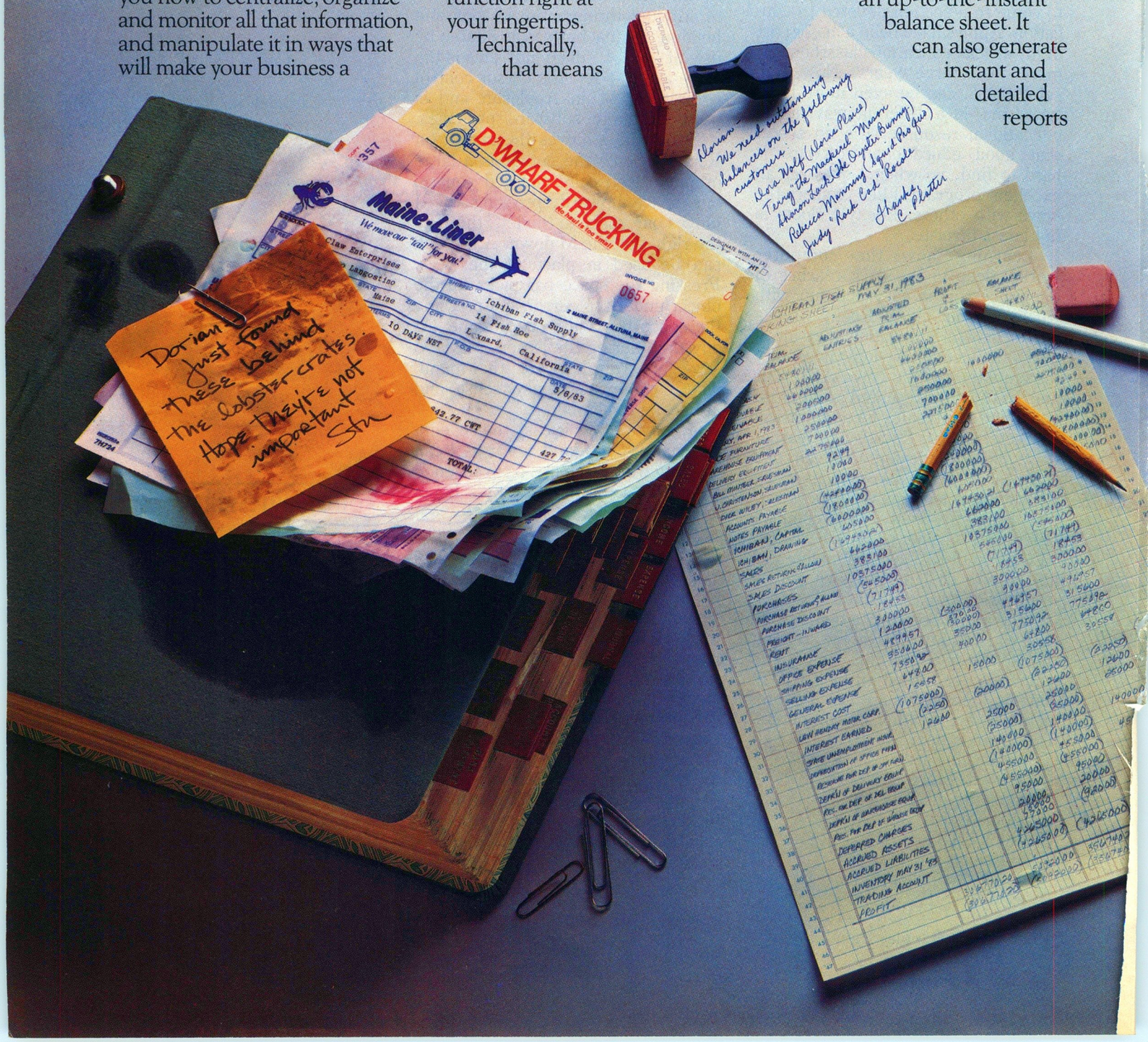
Attain instant financial status.

An Apple III, teamed with the BPI General Accounting Package, can put every basic accounting function right at your fingertips.

Technically, that means General Ledger, Accounts Payable, Accounts Receivable and Payroll—all in one package.

Meaningfully, that means you can turn numbers into answers.

With BPI, your Apple III can give you a snapshot of your company's financial condition, an up-to-the-instant balance sheet. It can also generate instant and detailed reports



gain your balance.

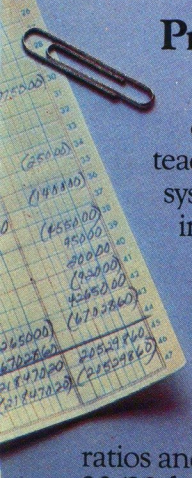
on your customers and vendors.

So you know who owes whom, how much, and how come.

And just how well your cash flow is flowing.

And where to give credit where credit is due (a customer inquiry feature allows you to make credit decisions based on the most current information).

You can also list your purchases by discount dates. And take advantage of them in no uncertain net terms. You can even keep payroll records without paying more, because it's part of the same package.



Profit from history.

In business as in life, experience is the best teacher. And the Apple/BPI system can provide you with instant comparisons of this-month-this-year vs. this-month-last-year, or this-year-to-date vs. last-year-to-date.

So you can quickly spot changing expense ratios and make decisions with 20/20 foresight.

Ichiban Fish Supply Company
Consolidated Income Statement
Current Comparative
Periods Ending May 31, 1983 and May 31, 1982

	May 31, 1983	%	May 31, 1982	%
Income				
Contract Sales	52,818.02	91.3	44,176.52	92.7
Retail Sales	5,016.88	8.7	3,500.00	7.3
Total Income	57,834.90	100.0	47,676.52	100.0
Cost of Sales				
Cost of Contract Sales	37,330.00	64.6	31,806.55	66.7
Cost Of Retail Sales	4,879.85	8.4	3,409.35	7.2
Total Cost of Sales	42,209.85	73.0	35,215.90	73.9
Gross Profit	15,625.05	27.0	12,460.62	26.1

Your Apple can generate instant income statements (with expense ratios) or balance sheets, and let you compare them to last month's or year's, then print them out to suit your banker.

The BPI General Accounting Package also lets your income statements be coded by location, department or product line. So you know where your money's coming from.

And where it's not.

Ichiban Fish Supply Company
Merchandise Purchased By Due Date
As Of 05/31/83

Date	Vendor No.	Name	Invoice Number	Acct No.	Detail	Net Amt
05/02/83	1	Herring World Due: 06/03/83	35278532	5010-01		501.23
05/05/83	2	Consolidated Cod Due: 06/05/83	4562	5010-01		209.36
05/05/83	3	Levy Sushi Farm Due: 06/05/83	212	5010-01		459.00
05/05/83	4	Mussel Men, Inc. Due: 06/05/83	657	5010-01		68.26
Total						1,237.85

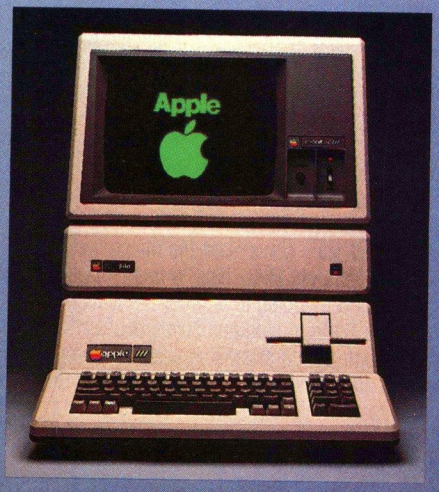
It can also allow you to take full advantage of merchandise discounts. So you'll know whom to pay, when to pay, how much to pay—and save a lot of clams in the process.

Make a timely statement.

Add an Apple Dot Matrix or Daisywheel printer to your Apple III, and you can print out your entire balance sheet in minutes.

Or any number of reports, from cash receipts to payroll ledger to income. You can even print checks and customer statements.

The impressively professional results will make an important statement to everyone you deal with—including your banker.



More ways Apples pay.

There are more people in more places doing more things with Apples than with any other personal computer in the world.

Ichiban Fish Supply Company
As of 05/31/83
Accounts Receivable Ledger
Page 1

Customer No	Name	Folio	Balance Forward	Current Month	Balance
1	Moser's Sole Food		892.79		
	Invoice 1124	IR		212.23	
	Invoice 1199	IR		156.56	
	Invoice 1326	IR		820.00	
	05/15/83	CR		250.00-	
					1,831.58

To avoid fishy transactions, you can instantly display customer's payments, charges and current balance. In this case, a few more cans of tuna would put Mr. Moser over his $2,000 limit.

Because for one thing, there's more software for Apples than for any other personal computer in the world. So the same Apple that handles all your accounting needs can also handle financial spreadsheets, word processing and electronic filing.

You'll also find programs that are designed specifically for your kind of business. Be it dentistry, architecture or swine herding.

Of course, the best way to learn all the ways Apples can help you make better business decisions is to visit any one of over 1500 authorized Apple dealers.

So drop in. For a full account.

apple®

The most personal computer.

Call **(800) 538-9696** for the location of the authorized Apple dealer nearest you, or for information regarding our National Account Program. Or write Apple Computer Inc., Advertising Department, 20525 Mariani Avenue, Cupertino, CA 95014. © 1983 Apple Computer Inc.

Circle 9 on Reader Service Card

Emulates these terminals exactly.

IBM 3101
DEC VT100, VT52
Data General D200
ADDS Regent 20, 25, 40
Hazeltine 1400, 1410, 1500
Lear Siegler ADM-3A, ADM-5
TeleVideo 910
Teletype Model 33 KSR

Apple is a trademark of Apple Computer, Inc.

SOFTERM®

High-speed CRT Look-alike Software for Your Apple®

New File Transfer Language

BREAK
CATALOG
CHAIN
CONFIGURE
CONNECT
CONVERSE
DIAL
END
HANGUP
LOG
MONITOR
NOLOG
ONERR
PAUSE
PROMPT
RECEIVE
REMARK
RETRIES
SEND
SPECIAL
SPEED
TIMEOUT
XMIT:WAIT

Supports these interface boards.

Apple Communications Card
Apple Parallel Printer
Apple Serial Interface
Apple Super Serial Card
Bit 3 Dual-Comm Plus™
CCS 7710,7720, 7728
Hayes Micromodem II™
Hayes Smartmodem™ 300, & 1200
Intra Computer PS10
Mountain Computer CPS Card™
Novation Apple-Cat II™ 300 & 1200
Orange Micro Grappler™
Prometheus VERSAcard™
SSM ASIO, APIO, AIO, AIO II™

Supports your 80-column hardware.

ALS Smarterm™
Bit 3 Full-View 80™
Computer Stop Omnivision™
M&R Sup'R'Terminal™
STB Systems STB-80™
Videx Videoterm™
Vista Computer Vision 80™
Wesper Micro Wizard 80™

Your host computer won't know the difference!

Softerm provides an *exact* terminal emulation for a wide range of CRT terminals which interface to a variety of host computer systems. Special function keys, sophisticated editing features, even local printer capabilities of the terminals emulated by Softerm are fully supported. Softerm operates with even the most discriminating host computer applications including video editors. And at speeds up to 9600 baud using either a direct connection or any standard modem.

Unmatched file transfer capability

Softerm offers file transfer methods flexible enough to match any host computer requirement. These include *character protocol* with user-definable terminator and acknowledge strings, block size, and character echo wait, and the intelligent *Softrans*™ protocol which provides reliable error-free transmission and reception of data. The character protocol provides maximum flexibility for text file transfers. *Any* type file may be transferred using the Softrans protocol which provides automatic binary encoding and decoding, block checking with error recovery, and data compression to enhance line utilization. A FORTRAN 77 source program is supplied with Softerm which is easily adaptable to any host computer to allow communications with Softerm using the Softrans protocol.

Softerm file transfer utilizes an easy to use *command language* which allows simple definition of even complex multiple-file transfers with handshaking. Twenty-three high-level commands include *DIAL, CATALOG, SEND, RECEIVE, ONERR, HANGUP, MONITOR* and others which may be executed in immediate command mode interactively or from a file transfer macro command file which has been previously entered and saved on disk.

Built-in utilities

Softerm disk utilities allow DOS commands such as CATALOG, INIT, RENAME, and DELETE to be executed allowing convenient file maintenance. Local file transfers allow files to be displayed, printed, or even copied to another file without exiting the Softerm program. Numerous editing options such as tab expansion and space compression are provided to allow easy reformatting of data to accommodate the variations in data formats used by host computers. Softerm supports automatic dialing in both terminal and file transfer modes. Dial utilities allow a *phone book* of frequently used numbers to be defined which are accessed by a user-assigned name and specify the serial interface parameters to be used.

Online Update Service

The Softronics Online Update Service is provided as an additional support service at no additional cost to Softerm users. Its purpose is to allow fast turnaround of Softerm program fixes for user-reported problems using the *automatic patch facility* included in Softerm as well as a convenient distribution method for additional terminal emulations and I/O drivers which become available. *User correspondence* can be electronically mailed to Softronics, and *user-contributed* keyboard macros, file transfer macros, and host adaptations of the Softrans FORTRAN 77 program are available on-line.

Most advanced communications software available

Just check Softerm's 300 page user manual. You simply can't buy a more sophisticated package or one that's easier to use. Available now for only $150 from your local dealer or Softronics, Inc.

SOFTRONICS

6626 Prince Edward, Memphis, TN 38119. 901-755-5006

Circle 37 on Reader Service Card

MAGIC SQUARE DANCE

J. Taylor

THE Magic Square is something most of us have encountered at one time or another, and has its origins in the earliest days of numeric literacy in the Middle East and India.

Essentially it is a series of integers starting at 1, placed in a cellular square in such a way that the sum of any side, vertical, horizontal or diagonal is represented by a constant.

The constant will be $n^3 + n /2$ where n is the number of cells per side.

The usual Magic Squares which have occupied attention are those with 4 or 5 sides (a 3x3 square is only capable of 1 solution), the constant for 4x4 square is 34, 5x5 is 65.

There are many hundreds of variations possible for 4x4, increasing dramatically for each unitary increment. What is so intriguing about these squares is that by joining the odd/even or all numbers in sequence, we get an interesting variety of geometric patterns.

All this is fairly well known and it seemed that the generation of these patterns to the High-Res screen might be an interesting exercise.

The program itself reads in as data a selection of some of these Magic Squares and places them to the text screen.

Switching to the High-Res screen, we plot the geometric patterns derived on three separate boxes.

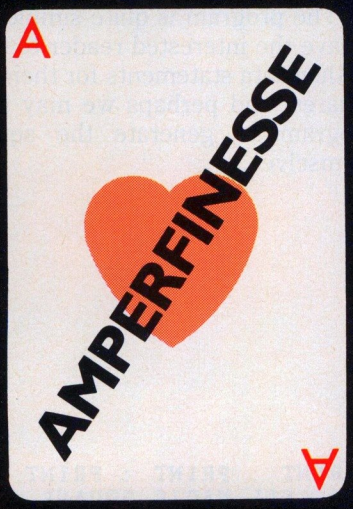
The program is quite simple, and I leave the interested reader to enter further data statements for their own squares, and perhaps we may see a program to generate the squares themselves.

```
10  REM

MAGIC SQUARE DANCE
   BY  J TAYLOR

 CALL -A.P.P.L.E.

15  HOME
20  DIM X(16),Y(16),A(16)
30 COUNT = 0
40  FOR I = 1 TO 16: READ X(I): NEXT

50  FOR I = 1 TO 16: READ Y(I): NEXT

60  FOR I = 1 TO 16: READ A(I): NEXT
    : REM  SQUARES DATA
70 HT = 1
80  HOME : TEXT
90  VTAB (1): HTAB (11): GOSUB 65
    0
100  HTAB (11): PRINT  CHR$ (255)
    ;: PRINT "MAGIC SQUARE NO.";
    COUNT + 1;: PRINT  CHR$ (255
    )
110  HTAB (11): GOSUB 650
120  VTAB (4): HTAB (12)
130  FOR S = 1 TO 9 STEP 2
140  VTAB (S + 4): HTAB (14)
150  GOSUB 640
160  NEXT
170  VTAB (6)
180  FOR I = 1 TO 16: REM  SQUARE
     TO TEXT SCREEN
190  HTAB (HT + 13): PRINT  CHR$
    (255);
200  HTAB (26): PRINT  CHR$ (255)
    ;
210 A$ =  STR$ (A(I)):A =  LEN (A
    $)
220  IF A = 2 THEN  HTAB (HT + 14
    ): PRINT A(I);: GOTO 240
230  IF A = 1 THEN  HTAB (HT + 15
    ): PRINT A(I);
240 HT = HT + 3
250  IF I = 4 OR I = 8 OR I = 12 THEN
    HT = 1: PRINT : PRINT
260  NEXT
270  PRINT : PRINT : PRINT : PRINT
    " IN A 4X4 CELL MAGIC SQUARE
     THE SUM OF"
280  PRINT : PRINT "   ANY SIDE,
    HORIZONTAL, VERTICAL OR "
290  PRINT : PRINT "            WI
    LL TOTAL 34."
300  HTAB (9): VTAB (24): INVERSE
    : PRINT " ANY KEY TO CONTINU
    E ";: NORMAL
310  VTAB (24): HTAB (31): GET Q$

320  HTAB (1): VTAB (21): PRINT "
       PLOT ALL    PLOT  EVEN
     PLOT  ODD"
330  PRINT "   -NUMBERS------NUMB
    ERS------NUMBERS-"
340  HGR : HCOLOR= 3
350  HPLOT 10,40 TO 90,40 TO 90,1
    20 TO 10,120 TO 10,40
360  HPLOT 100,40 TO 180,40 TO 18
    0,120 TO 100,120 TO 100,40
370  HPLOT 190,40 TO 270,40 TO 27
    0,120 TO 190,120 TO 190,40
380  FOR I = 1 TO 16: HPLOT X(I),
    Y(I): HPLOT X(I) + 90,Y(I): HPLOT
    X(I) + 180,Y(I): NEXT
390  FOR I = 1 TO 16: REM  PLOT A
    LL
400  IF A(I) = 1 THEN  HPLOT X(I)
    ,Y(I): GOTO 420
410  NEXT
420  FOR L = 2 TO 16
430  FOR J = 1 TO 16
440  IF A(J) = L THEN  HPLOT  TO
    X(J),Y(J)
450  NEXT J: NEXT L
460  FOR I = 1 TO 16: REM  PLOT E
    VEN
470  IF A(I) = 2 THEN  HPLOT X(I)
     + 90,Y(I): GOTO 490
480  NEXT
490  FOR L = 4 TO 16 STEP 2
500  FOR J = 1 TO 16
510  IF A(J) = L THEN  HPLOT  TO
    X(J) + 90,Y(J)
520  NEXT J: NEXT L
```

AVANT GARDE

Presents

More Magic!



FORTRAN-like Applesoft enhancement for your Apple

- *Amperfinesse* brings the power of such high level languages as Fortran and PL/I to the Apple II Plus at an inexpensive price.
- Plus a complete set of editing commands!
- Call subroutines with arguments and by name!
- GOTOs by name!
- Formatted numeric output including '$' and '*' options!
- Input virtually any character string!
- Search part or all of an Applesoft program for a string!
- Substitute one string for another anywhere in a program!
- Renumber part or all of a program!
- Initialize a disk without DOS and add 10000 bytes additional storage!
- Add simple music and sounds to a program with easy to use music commands!
- Initialize the hi-res screen to any HCOLOR and set ROT, SCALE, and HCOLOR, all with a single command!
- Auto-line numbering mode!
- Move an Applesoft program almost anywhere in memory, even while the program is running!
- Redimension arrays
- and MORE!

NEW

from

AVANT GARDE

P.O. Box 30160/Eugene, OR 97403

(503) 345-3043

Circle 10 on Reader Service Card

MAGIC SQ. DANCE

```
530   FOR I = 1 TO 16: REM  PLOT O
      DD
540   IF A(I) = 1 THEN  HPLOT X(I)
       + 180,Y(I): GOTO 560
550   NEXT
560   FOR L = 3 TO 16 STEP 2
570   FOR J = 1 TO 16
580   IF A(J) = L THEN  HPLOT  TO
      X(J) + 180,Y(J)
590   NEXT J: NEXT L
600   VTAB (24): HTAB (30): GET Q$

610 COUNT = COUNT + 1
620   IF COUNT = 8 THEN  END
630   GOTO 60
640   FOR I = 1 TO 13: PRINT  CHR$
      (255);: NEXT : RETURN
650   FOR I = 1 TO 19: PRINT  CHR$
      (255);: NEXT : PRINT : RETURN

660   END
670   DATA   20,40,60,80,20,40,60,
      80,20,40,60,80,20,40,60,80,5
      0,50,50,50,70,70,70,70,90,90
      ,90,90,110,110,110,110: REM
       X,Y COORDINATES FOR HI-RES
680   DATA    10,11,8,5,6,7,12,9,1
      5,14,1,4,3,2,13,16: REM
      DATA FOR SELECTION OF MAGIC
      SQUARES
690   DATA   16,3,2,13,5,10,11,8,9
      ,6,7,12,4,15,14,1
700   DATA  7,12,1,14,2,13,8,11,16
      ,3,10,5,9,6,15,4
710   DATA    1,8,13,12,14,11,2,7,
      4,5,16,9,15,10,3,6
720   DATA   16,2,3,13,5,11,10,8,9
      ,7,6,12,4,14,15,1
730   DATA  7,12,9,6,14,1,4,15,2,1
      3,16,3,11,8,5,10
740   DATA   15,10,8,1,6,3,13,12,9
      ,16,2,7,4,5,11,14
750   DATA   3,13,12,6,16,2,7,9,5,
      11,14,4,10,8,1,15
760   REM
J.J.TAYLOR 14 WOODWAY CLOSE,
TEIGNMOUTH DEVON ENGLAND
```

```
]LIST

10  REM

  SO WHAT DID YOU EXPECT DEPT.

VAL J GOLDING : CALL -A.P.P.L.E.

100   LOMEM: 3072
110 A% = 21327:B% = 8279:C% = 184
      97:D% = 21536:E% = 17481:F% =
      17440:G% = 22863:H% = 21792:
      I% = 17752:J% = 20549:K% = 1
      7236:L% = 8255
200   FOR I = 3074 TO 3152:A$ = A$
       +  CHR$ ( PEEK (I)) +  CHR$
      ( PEEK (I + 1)):I = I + 6: NEXT

210   HOME : VTAB 8: HTAB 20 - ( LEN
      (A$) / 2): PRINT A$
```

The Avant-Garde Programmer's Series:

Better Than Magic!

HI-RES ELECTRONIC DESIGN

Create and print electronic circuit plans. Keyboard selection and rotation of 98 different electronic components. Paddles or joysticks provide easy placement. **$29.95**

HI-RES SECRETS

Complete Apple® graphics utility fills the needs of today's creative programmer. 263 pages of revealing documentation. Completely relocatable color or graphic, animation and sound routines. The most useful package on the market. **$125.00.**

Demo available $10.00

HI-RES ARCHITECTURAL DESIGN

Plan and design individual rooms, complete floor plans and total buildings. 75 different floor plan shapes can be easily rotated and moved. Lengths, diagonals and angles are calculated on-screen. **$29.95**

SUPER QUALITY EPSON HI-RES DUMP

Best quality Epson dump features true black copy. Mirror, negative, flip or rotated images available in three sizes. Automatic horizontal centering. **$25.00**

HI-RES SECRETS GRAPHICS APPLICATIONS SYSTEMS

Step from Basic to better Basic programs, Basic to Assembly hi-res graphics. Create graphs, electronic and architectural designs, arcade and adventure games, more! Fastest color-fill; 300 colors/patterns. **$75.00**

PAINT MASTER SCENE UTILITY

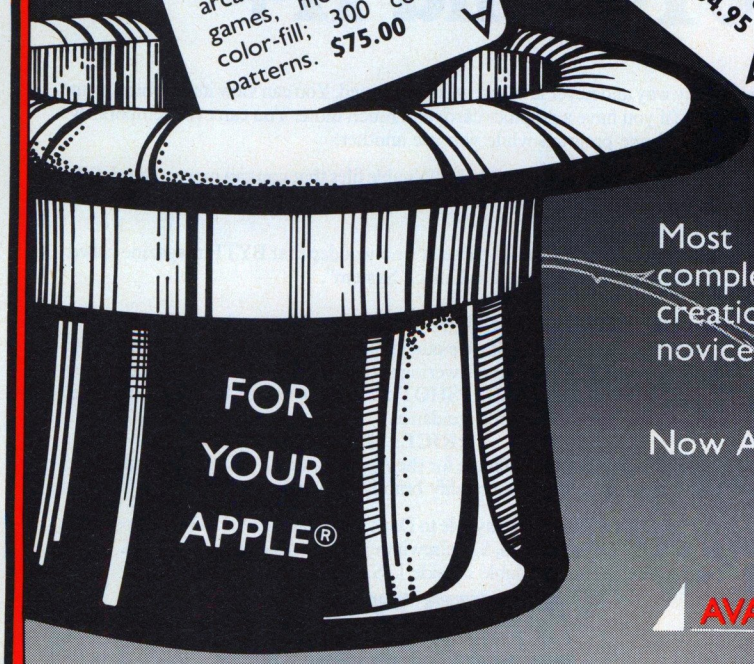
Allows line drawing, scene painting, screen scene editing, compression. Fastest color-fill routines anywhere. Perfect for adventure game creation or computer art. Over 300 colors/patterns. use creations in your own program. **$34.95**

SUPER SHAPE DRAW AND ANIMATE

Draw in any direction using keyboard commands. Two types of animation, rotation, scale, background color, shape color and more. **$34.95**

ACTION SOUNDS AND HI-RES SCROLLING

Machine language sounds, hi-res scrolling routines and Super-font program in one useful package. **$18.95**



Why Better?

Because There Are No Secrets!

Most of these programs are unprotected or completely listable and may be used in all your creations, with no royalties required. They turn novices into professionals and professionals into absolute magicians!

Now Available from Your Local Software Dealer and by calling us directly at

(503) 345-3043

AVANT-GARDE CREATIONS, INC.

P.O. Box 30160 Eugene, OR 97403

ARE YOUR CREATIONS MARKETABLE? ARE YOU A QUALIFIED PROGRAMMER LOOKING FOR ASSIGNMENT? GIVE US A CALL.

Only one copy system lets you backup 64k programs in 20 seconds flat...

...the Snapshot Copykit™ for Apple IIe and II+

While other manufacturers have been making wild promises, we've been quietly developing the superior copy system you've been waiting for: the SNAPSHOT COPYKIT.

THE FLEXIBLE CARD – The SNAPSHOT COPYKIT is a board for all systems. It sits in any slot on the Apple IIe, II+ and II as well as software-compatible machines like the Basis 108 and the Franklin Ace. So, you can upgrade your system without paying for another card.

Although **NO LANGUAGE CARD/16K CARD IS REQUIRED,** the SNAPSHOT COPYKIT can be used with any memory card and all peripheral cards.

The SNAPSHOT COPYKIT is the only copy card that stays up-to-date. Because all its software is in RAM, rather than inflexible ROM, the new enhancements we continyally develop are made available to you at nominal cost — immediately.

THE USER-FRIENDLY CARD – We refined backup techniques for two years to make the SNAPSHOT COPYKIT just right for you, whether you're new to the Apple or a seasoned pro.

Other copy cards force you to use complex, unreliable procedures and repeated booting of 'utility disks' to copy programs larger than 48K. Outdated 'nibble copier' programs involve frustrating trial-and-error parameter changes. Only the SNAPSHOT COPYKIT backs up all 64K memory-resident programs (and most larger programs too) with one simple menu and one press of the trigger.

THE POWERFUL CARD – The SNAPSHOT COPYKIT gives you back the power over your computer that copy-protected software took away. You can interrupt a running program, copy it, list it (even if it's 'unlistable'), disassemble it, step and trace it, modify it and resume running it. The program is not disturbed in any way and never knows it was interrupted. You can view its graphics, print them (if you have a graphics card) and much more. You can even temporarily suspend one program while you use another.

All backups are automatically BRUNnable files that you can move to hard disk. Our state-of-the-art compression facility lets you put a number of backups on one floppy.

With powerful features like these, it's no wonder that BYTE magazine called SNAPSHOT "The Ultimate Unlock System".

The SNAPSHOT COPYKIT has many more features for programmers, gamesters and others. Write for our info packet or leave your address and questions with our 24-hour answering service at (413) 584-7600. We regret we cannot return all calls. All SNAPSHOT owners receive out 'Technical Hot-Line' number for instant technical guidance.

PRICE: $139.95

(Snapshot Two is still available for the Apple II+ and Franklin Ace at $119.95. 16K Card required; please specify brand when ordering).

TO ORDER: Mail check payable to Dark Star Systems. Add $3 for shipping in the U.S. ($7.50 to Canada, $15 elsewhere). Mass. residents add 5% tax. COD costs $5 extra (not available outside U.S.).

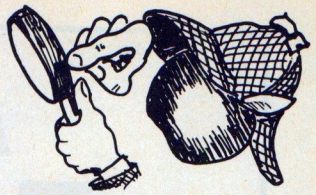
dark Star SYSTEMS

P.O. BOX 140, Amherst, MA 01004

Apple is a reg. trademark of Apple Computer.

NORTHERN SPY:

NELLIE AND THE PIRATES



Rick Sutcliffe/Call —A.P.P.L.E. staff writer

"HEY, sir, look at this!"

That is one thing that you can rely on about Nellie Hacker. She certainly does not stand on ceremony. When she has something to say — you can be right in the middle of an important task of your own (like reading the cartoons in *Leadership* magazine, or marking a batch of mediocre Physics 12 labs) — and she just bursts right in, destroying your concentration, laugh track and all.

"I thought you were typing up one of my *Call -A.P.P.L.E.* articles tonight." I shot back to the other end of my lab where Nellie sat hunched over the computer keyboard. (She has been my secretary for the last few months.)

"Oh, don't worry, I keep track of my time. You get your money's worth out of this slave labour, and besides, I'm finished, so I thought I would boot up this disk I got from Alex the other day. It looks like a giant accountant's worksheet with rows and columns that you can put all sorts of formulas into. This is really nifty!"

Nellie was gushing again, but that's nothing new, and I'm used to it by now.

"Sounds like a spreadsheet to me, Nellie. Where did you buy it and how much did you pay?" I solicited.

"Oh, Alex just ran me off a copy and photocopied the manual he was using; he didn't charge me anything."

I had finished marking the last of the labs by now — a good one too, and wouldn't you know it was hers — so I decided to return it there and then and to see what she had. At my request, she re-booted the program, and there in the upper left hand corner was the logo I was half expecting.

"You've got a ripoff of Busy-Calc, Nellie." I tried to make it sound like bad news.

"What's that?"

"It's a well known program of this type. They have sold thousands of copies on protected disks that you are not supposed to be able to duplicate. It retails for about $250."

"You mean..."

"Yes, Nellie. You've got a hot program in your hot little hands; why, I could turn you in — some manufacturers give pretty big rewards."

"But, how could I know? There's no copyright notice in the manual; it's just a photocopy of a typed looseleaf, and there aren't any trade names or anything." Nellie was indignant. "Wait until I see that Alex."

"This one I know because I bought it myself," I responded, trying to be soothing. "Obviously, some enterprising soul has produced this brief manual in an attempt to make money from another person's program.

"Besides, Alex could never in his lifetime come up with a program like this, and you can bet there's nothing of this quality in the public domain either."

"That's the stuff anybody can copy, right?"

"Right, Nellie."

"But, who can afford to pay retail for this? I mean, like $250 is pretty ridiculous. If you own a business or something, you pay the shot of course. But, what about us students? It's one thing to pay, maybe, thirty bucks like we would for a text, but this kind of price is just out of sight!"

"Well, Nellie, the manufacturer would say that he needs that high price to justify the cost of production."

"That's absolutely ridiculous. A disk costs about two dollars, the manual no more than eight, right, and even if they pay that much out in overhead and split the same with the author as profit — it still comes to only thirty dollars."

"Actually, advertising is pretty expensive, and they might have to pay the author of the manual — after all, a lot of programmers are illiterate." She gave me a savage look, so I quickly added; "Not you, of course, your writing is as brilliant as always. Here, have your lab back," I added, hoping that she would be mollified by yet another perfect score.

"Furthermore," I continued without missing a beat, lest we start a new conversation before this one was half done, "even if it went to a wholesaler for, say, sixty dollars, he might as much as double the price, and the retailer could do the same. Add a few taxes along the way and you've got your $250 in jig time.

"Substantial discounts are usually available in the United States, but not up here. Why, when I bought the word processor that you use to do my articles, it listed for US $130; I paid US $80 and a week later spotted it in a Vancouver store for CDN $225. Even with taxes, duty and exchange added in, the R/O factor is still more than two. Hey, kid, people are getting rich in this business."

"I still say the whole thing is absurd. Let's set aside wholesale and retail ripoff pricing policies for a minute and look at it this way: Suppose as an author you are getting a twenty percent royalty. Now, wouldn't you rather have a fifth of 20,000 copies times a selling price of $20 than a fifth of 500 copies times a price of $250? After all, if you offer a good product for a sensible price, aren't you going to sell it by the bag full?"

Nellie was not the sort of person who needed to waste much time on mental multiplication, and I could see that she obviously understood the business principles involved better than I had thought, so I replied: "I figure it that way too, girl, and so do a few software houses that are cutting prices from the $200-$300 range to a tenth of that. In fact, some of them have ended up backordered by thousands of copies as a result. Actually, that can destroy your business if you are not careful, but perhaps it is a better type of problem than sitting around waiting for the occasional order to come in.

APPLE® MODEM. WITH EVERYTHING.

Your best buy in modem history. The **Networker**™ a plug-in single-slot direct connect modem for your Apple II, II+, and IIe (or Franklin, Albert, or other Apple compatible computer). You are immediately linked to the vast network of computers — data bases like the Source,® Compuserve,® or Dow Jones,® friends, banks, businesses, and hundreds of local "bulletin boards." For only $129. There's absolutely nothing else to buy. And it even comes with software.

$129

This is the modem that does it all, and does it for less. The Apple Communications Card is on board, so you won't need to buy one. That will save a bunch. It's 300 baud, the most commonly used type of modem. It comes complete with its own communications software on Apple compatible disk, giving you features ***no*** modem offers.

Like "data capture" to lock the messages into your Apples' RAM, and then move the information onto a disk for easy reference and review. A terminal program that turns your computer into a communications command center, displaying on-screen "help" menus, continuous updates of memory usage, carrier presence, baud rate, communications status, all while taking in and displaying information from any computer.

The **Networker** supports both originate and answer modes, so you can send and receive information, in full or half duplex modes. And its all easily programmable right from the keyboard of your Apple computer. All you do is plug **Networker** into any slot (except 0) in your Apple computer. It'll even ask you what slot it's in. It's that easy to use.

NETMASTER™ COMMUNICATIONS SOFTWARE

For $179 we include the even more incredible **Netmaster** software for advanced users. **Netmaster** is a "communications freeway" that lets you speed any Apple file to another Apple through noisy phone lines, without errors. **Netmaster** even includes its own Disk Operating System to control the flow of information and leave you over 40K of "buffer" memory from your 64K RAM. And there's always help when you want it. On screen. Status indicators keep you fully updated.

Netmaster will let you transfer games, computer graphics, programs, sales reports, documents, any DOS 3.3 file, all directly from disk to disk, all with computer (not telephone) accuracy. **Netmaster** is completely configurable to understand any computer, even mainframes. File size is unlimited.

But **Netmaster** is not stuffy. It will talk to the other communications packages like Visiterm® or ASCII Express.® Of course they don't support **Netmaster's** superb error checking, that guarantees you won't miss even a comma. It doesn't only talk to the Networker either. It will also talk to those more expensive modems from Hayes and Novation.

And **Netmaster** is fast. It transfers information disk to disk three to five times faster than the others. Of course **Netmaster** is fully compatible with them. But if you want things to happen really fast, talk to another Netmaster.

We sell **Netmaster** by itself for $79, and even if you have another modem for your Apple, **Netmaster** is an outstanding value. It will even support auto dial and auto answer. **Netmaster** requires 48K of Apple memory, disk drive and DOS 3.3, and ZOOM's **Networker** modem or another modem. It can also be used without a modem when two Apples are hardwired together, using a 6850 or 6551-based interface card.

Networker, **Netmaster**, and **Networker/Netmaster** combo are all produced in the USA by ZOOM Telephonics, producers of the popular **DEMON DIALER.** The Networker is FCC registered, and all products come with a one-year warranty.

HOW DO I GET ONE? OR TWO?

Check with your computer dealer. Or call us direct at 1-800-631-3116, and we'll let you know where to get them in your area.

Massachusetts residents call 617-423-1288.

The following are trademarks: Apple by Apple Computer, the Source servicemark by Source Telecomputing, Compuserve by Compuserve, Dow Jones by Dow Jones & Company, Visiterm by Visicorp, ASCII Express by Southwestern Data Systems, DEMON DIALER by ZOOM Telephonics. 1983 ZOOM Telephonics, Inc.

ZOOM

ZOOM Telephonics, Inc.
207 South Street, Boston, MA 02111

Circle 37 on Reader Service Card

NELLIE AND THE PIRATES

"However, it seems that most of the people who have been in the business for a few years have grown up with the fat prices and are sticking to those high markups with the low volumes that they imply. I guess you could call them traditionalists — if you can use that word in an industry which has only been around for about seven years."

"I hope they go broke." Nellie put in.

"So do I, Nellie, so do I."

She gave a sort of "harrumph," that sounded suspiciously like one of my own, as she dumped the offending "document" into the round file along with the label from her diskette. The latter quickly went back in her box of blanks. I'm proud to say, that girl does have a well-tuned sense of what's right.

"By the way, I thought you said that these kinds of diskettes were uncopiable," Nellie proffered after a few moments. "How did Alex get one that he could copy for me?"

"Oh, Nellie, there is no such thing as a really uncopiable disk. There are all sorts of programs around that are designed to do just that. In fact, what with one group of programmers inventing new ways to lock diskettes, and another group working on ones to copy them despite the locks, a tremendous amount of time and talent has been tied up in the last few years on this problem alone. You should read the magazine ads more often."

"Haven't the time," she sulked. "You should see the English assignments Mr. M is giving us."

"And I'll bet," she added after the briefest of pauses, putting us abruptly back on the subject, "that they charge plenty for their picklock software and that their programs can't copy themselves."

"Of course they can't," I responded, answering her points in reverse order, "and they do charge between fifty and a hundred dollars, but people just buy two different kinds of copiers. In fact for that higher price, you can pick from several kinds of hardware devices which work by making a 'snapshot' of everything in memory and then putting it on a disk in an unlocked format. For those, you don't even need any software, it's all built into the hardware card."

"That's disgusting." Nellie was becoming quite agitated, as she always does when the issue under discussion offends her moral sensibilities. "If they really thought they were providing any kind of legitimate service, they would sell their own stuff for a lot less and on copiable disks at that. They're no better than all the rest of them — just out to cheat us of all the money the traffic will bear."

"Now, hold on a minute, kid. What if you do buy one of those expensive locked disks, because it just happens to be the best program available for your needs (and you also have the money to do it). Do you really want to get stuck without a backup? Surely Mr. H. and I have trained you better than that.

"After all," I continued, "some manufacturers are terrible when it comes to replacements for a damaged disk. If they do it at all, you have to deal through the mail, and that can take forever. What's worse, in the year or more that it might take for the disk to go bad, they might even be long since broke and out of business.

"Only a very few include a backup as part of the original package; one manufacturer I bought from was trying so hard to save money that he put the spare copy on the back of the original diskette, can you believe it? I wonder if he thought people could somehow send back only the damaged side and keep the good side operating until the replacement arrived.

"Wouldn't you rather be able to make your own cheap copies for backup as soon as you buy a program, just in case? Wouldn't you like the freedom to have several copies of a word processor configured for your different writing needs? Wouldn't you also like to be able to save some three hour long game on a disk so that you can resume play at a later date?"

"Uck, you know I can't stand all those gamey things. They are full of nothing but violence and fantasy. Give me concrete applications for computers any day over silly games.

"Besides, is even that sort of copying allowed? I've read some of the package notices and they can make it sound like you're a criminal if you even read the thing from the disk into the memory of the machine."

"I must have told you a million times, Nellie, don't exaggerate. But yes, it is allowed. As long as you do not sell or give the copies away, you can make as many as you want for your own use. What the vendors are most worried about is the large scale ripoff artists who run off hundreds of copies and sell them cheap or give them away at their club meetings.

"The worst offenders are the schools. A lot of Districts buy a single copy of some major software package and then proceed to run off one for each machine in their jurisdiction. That can run into the thousands in some cases.

"The trouble is, in their zeal to stop these big thefts, the manufacturers in their turn are interfering with the legitimate rights of their honest customers. Why some of them include the most extravagant anti-copying and warranty disclaimer notices I have ever seen on any product. Sometimes they do not even realize that there is no support in law for much of what they say in these statements, and that they are entirely null and void if not placed on the outside of the package where they can be read before purchase. They try to be as intimidating as possible to protect their investment, I suppose."

"But, it's like I keep saying, sir. If they are going to ask outrageous prices like $250, nobody is going to buy more than one copy, and precious few are even going to do that.

"They are just making it too tempting to be a crook. Can't the people who steal software in that way see that that is just what they are — crooks? Not only that, but that they are ruining the market for the people who want to write good programs, especially for the schools — if they are as bad as you say in this respect."

She continued her scattergun condemnation of all involved, both indignant and a little breathless: "Besides, what are they supposed to be teaching us kids about honesty, and trust and all that stuff?." (Like most brilliant young people with something to say, Nellie was a lot more careless in her spoken English than in her writing.)

NELLIE & PIRATES

"I'm glad to hear that you care too, Nellie, but let's be realistic: What fraction of the people in our society really do? As you say, the prices are far too high — they do make the temptation too great for most people to handle. Not only that, honesty in such matters is not even an issue with most people. They might not hold up a store; they might even hesitate about shoplifting, but if they can get away with stealing something in the privacy of their homes, they do not even give it a second thought. It takes a higher power than the law of the land to change people, so the problem is not going to be solved just by passing tougher laws.

"Tell you what, kid, suppose we tackle it from the other end by looking at what can be done about the price problem. I see a lot of software — you have typed some of my reviews, and I have done a lot more short ones myself at home - so I don't need to tell you what I think about the quality of software being offered for those high prices — especially the so-called educational packages. I have come to the conclusion that there isn't a program available today for any micro that can justify even a $100 price tag at retail.

"Games should be under twenty dollars, utilities up to thirty, spreadsheets and word processors up to sixty, and systems packages at the upper limit of a hundred.

"You should have unlocked software and be able to make as many copies as you want for yourself, with manuals selling for 25% of the package cost on the understanding that an organization can place the program at one location per manual they buy.

"If prices were brought within reason, people would willingly foot the bill for the originals and piracy would not pay. Then the better software availability and lower prices would result in higher sales volumes for both programs and machines, and we would go into a spiral cycle of higher demand coupled with lower costs due to economies of scale. Ultimately the whole industry would benefit — we poor consumers most of all."

"That would help the individual owner, and schools too, but how would you deal with the clubs and others who just pass the stuff around to their friends for free?" Nellie asked.

"The same way as they do it in the music industry, Nellie."

"I can see how you could put a royalty on blank disks as they do on cassette tapes and then divvy it up among the programmers, but how do you decide on each person's fair share? You can't very well monitor the number of times a program is run," she added, with a look of puzzlement. "After all, there isn't any such thing as air time in the computer industry, you know."

"I wasn't talking about air time, Nellie. What I had in mind was just collecting the sales figures on the various commercial programs from the software houses. This could be done by a central agency which could then divide up the royalties to authors and publishers on a percentage basis calculated on their share of the total amount. They would go along with it because it would mean more revenue, and the disk manufacturers would too, because in the long run, the general volume increases would result in them also selling more of their products."

"It sounds like a good idea, but do you really think they will take the first step by lowering prices and taking off the copy protection?"

"Nellie, girl, the ones who do are going to make money faster than they can count it, and the ones that do not are going to be watching from the bankruptcy courts."

"That seems fitting." Nellie's sense of justice seemed to be satisfied.

"It does, doesn't it?" I replied, and then added: "Oh, and Nellie, before I forget, when you see Alex tomorrow in Mr. H's class — tell him I would like to have a little chat with him in my office at noon."

"O.K., but you aren't going to yell at him too much are you?" she solicited in a concerned tone — her earlier irritation now evaporated. (Nellie never stays even a little angry for long.)

"No, Nellie, you know me better than that. Let's just say that I'm going to have another go at educating the boy. He's a valuable young man; he just has some growing up to do."

"This is changing the subject, sir, but your mention of bankruptcy made me think of poverty, and I began to wonder. How do you really feel about quitting teaching here at the High School after all these years? I know that the College where you are going won't pay you nearly as well. Do you have any second thoughts?"

"It's only been twelve years — as long as you have been in school yourself, girl. And no, I am quite sure that the time and the place are both right. Besides, you're graduating and going on to University this year, so just think of me as doing the same. I must admit that it will be hard in some ways, for there are many good memories here, but some of you are graduating this year anyway, and a few will be going with me to Trinity Western College, so I won't be lonely."

"They have Apples there too, don't they, sir?"

"They sure do, Nellie, so I'll be right at home. If we can attract the students, we will be able to set up a degree program in Computer Science and, of course, we would just keep right on using them. Micros are the wave of the future; I am convinced that the people who are investing in timesharing minicomputers are going to have some very costly surplus on their hands in about three years time."

"Oh, I don't think that you will have any difficulty getting the students.' Nellie responded, picking out from what I said only the part that mattered most to her. "Why don't you do up a column on Trinity sometime as The Northern Spy and then see if you can persuade Kathryn to run it in the educational column in *Call -A.P.P.L.E.* There can't be many places like TWC, and it would sure be good publicity."

"Why, that is a great idea, Nellie — how about two columns from this one, when I've finished those workalike reviews? That might just fit in well with *Call -A.P.P.L.E.'s* plans.

"Going home early?," I added as she started to pack up her books.

"Early?" she expostulated, "It's after five, sir."

And so it was. Time to pack it in for the day.

News

As you will recall, we discussed the Apple Computer Corporation's marketing difficulties in this space a few months ago. At this writing, our predictions are looking pretty good. (That is, for the prophet. Whether Apple views it that way or not is another matter.) They are still (mid May) nowhere near caught up to orders as yet, and it will be at least midsummer before anyone in this part of the country will be able to walk into a dealer, order a *//e* and hope to get it say, within a week. Backlog at the higher volume stores is still in excess of six weeks, and orders are still coming in at least as fast as are the machines.

Apple has indeed continued to make the plus as we predicted, but only for dealers ordering in bulk — say several hundred machines at a time. If you can get one, they now sell for about the same as some of the "fake Apples" were going for last year.

There should be news on new introductions soon, if not by the time you read this. We should be looking for the mid-range Mackintosh, a low-end machine, new drives and DOS (SOS) for the *//e*, and a great outpouring of new (and cheaper) software and hardware for the *//e* from a multitude of sources.

I now think it most likely that we shall see a ///+ brought in — Apple does not abandon its babies — say at the present *//e* price. Maybe, just maybe, the *//e* will drop to under $700. The Mackintosh will inhabit what has been the ///'s price territory. Bold? Yes, I would say so. However, boldness is the name of the game, and Apple has never been lacking in it.

The *//e* Corner

As part of research for a later column on the workalikes and clones, I had my very own *//e* for three weeks recently. It is indeed all that it has been touted as and more — a full comparison report begins next month. (DV)

Two tidbytes now though. First, I read recently that the *//e* uses zero page locations $1F and $08. It ain't so, folks. It only uses $1F, and this stores the Y register before jumping into the monitor extension, and not all the time at that. You may get a conflict, but it is not that likely. I strongly recommend the new reference manual, despite the very high price. You can correct this and other errors you see in the print media. (Several magazines have gotten the locations or the functions of the new hardware switches very, very wrong.)

Second, the proper way to turn off the new 80-column card is *not* just to type IN#0 when the card is fully active. Instead, first send through a [Ctrl-O] to drop into active-40 mode, then issue the IN#0. You can do this from a BASIC program by PRINT CHR$(17) + CHR$(4) "IN#0".

Here's another way to do it, this time from machine language:

```
HKSET     EQU $A851
ETYON     EQU $C01E ;0D if off,
          $8D if on
ACTVKILL  EQU $CDAA
SIGBYTE   EQU $FBB3 ;$06 if //e,
          $38 if two, $EA if plus

*****************

          ORG anywhere
          ;relocatable
TWOEOFF   LDA ETYON
          ORA SIGBYTE
          EOR #$8F ;ZERO
          NOW IF //e
          BNE NOTON ;NOPE
          JSR HKSET ;YES,
          MAKE DOS HAPPY
          JMP ACTVKILL
          ;AND DO IT
NOTON     RTS

******************
```

Remember, to turn off the eighty-column card from command level or BASIC, you do issue the DOS command IN#0, but only from active-40 mode, not from active-80 mode. You're welcome, and I'll see you next time?

Attache-style cases for carrying and protecting your complete computer set-up. Accommodates equipment in a fully operational configuration. Never a need to remove equipment from case. Simply remove lid, connect power, and operate.

AP101	Apple II with Single Drive	$109
AP102	Apple II with Two Disk Drives	119
AP103	Apple II, 9 Inch Monitor & Two Drives	129
AP104	Apple III, Two Drives & Silentype Printer	139
AP105	13" Monitor with Accessories	99
AP106	AMDEK Color Monitor	119
RS201	TRS-80 Model I, Expansion Unit & Drives	109
RS204	TRS-80 Model III	129
AT301	ATARI Computers with Peripherals	109
P402	Centronics 730/737 & Radio Shack Printer	89
P403	Epson MX70/80 or Microline 82A	89
P404	Epson MX100 Printer	99
P405	IDS 560 or Prism 132 Printer	109
P406	Starwriter/Printmaster F-10 Printer	119
P407	Okidata Microline 83A or 84 Printer	99
P408	Prowriter 2 Printer	99
P409	Prowriter (Apple Dot Matrix) Printer	89
IB501	IBM Personal Computer	129
IB502	IBM Monitor	99
HP601	HP41 with Accessories	99
CM703	Commodore Model 64 with Drives	119
CM704	Commodore Model 64 with Dataset	109
NS010	North Star Advantage	139
CC80	Matching Attache Case (5")	85
CC90	Matching Attache Case (3")	75
CC91	Matching Accessory Case	95
CC92	5.25" Diskette Case	49

computer case company

5650 Indian Mound Court
Columbus, Ohio 43213
(614) 868-9464

CALL TOLL FREE
800-848-7548

Circle 7 on Reader Service Card

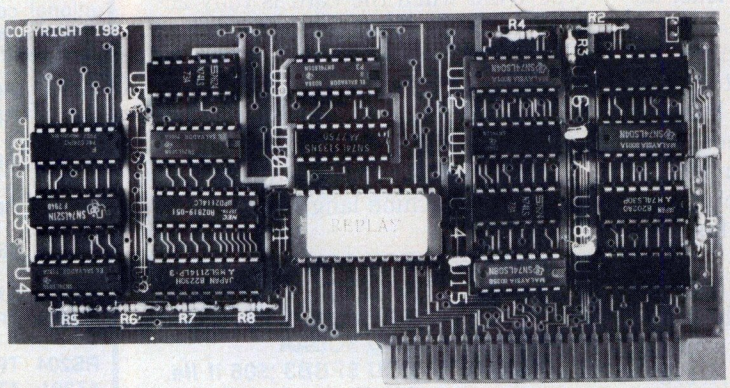
REPLAY II

Apple Program Copy And Development System

Now for IIe Also

- **Disk Formatting Irrelevant**
- **Does not interfere with other cards**
- **Card is transparent until copy is desired**
- **Copies all 64K with *no* compression**
- **Menu driven**
- **Copy and restart in under 15 seconds**
- **Copies memory resident programs**
- **Analyze programs**
- **Development tool**
- **Transparent step & trace**
- **Backwards & forward disassembly scroll**

Reviewed in Hardcore Magazine
Softkey Publishing
"The Replay card is very easy to use and understand... of all the cards I have used, I have found the Replay II to be the most powerful card available."



Why ours is better!

1. Accurate copy of memory
2. Language card not needed to copy or run - when using Replay Card
3. Fast copy & restart - 15 seconds -
4. Analysis programs and 60 pages of documentation
5. Copy system in eprom - No need to boot other disks at copy time -

To order or request info:
Write or call
Micro-Analyst, Inc.
P.O. Box 15003
Austin, TX 78761
(512) 926-4527
Dealer Inquiries Welcome

Cost:
Replay II or IIe $130
Please specify when ordering.
Machine language eprom $20
Step and Trace

If both ordered subtract $5.
Outside U.S./Canada add $10 shipping.
Texas Residents add 5% sales tax.
VISA/MASTER CARD accepted!!

REPLAY II is intended to be used as an analysis tool, for program development, and for making archival backup copies.

*Apple is a registered trademark for Apple Computers Inc.

REPLAY II is an interface card that is slot independent. Users can stop a program, examine and change memory, or copy the program, and restart. Control of the APPLE is obtained by pressing the remote switch which comes on an 18 inch cord outside the APPLE. REPLAY II does not copy the original disk, rather it copies the program executing in memory. If a copy is desired a blank disk is inserted in drive 1 and the options on the menu are contained in the eprom on the REPLAY II card, no other disk needs to be booted for copying, unlike other copy cards. The very act of booting another disk alters memory which is detectable by some protected software.

REPLAY II does not change ANY memory. Extra memory is buffered to allow copying and analysis without altering the original memory contents. Other copy cards always change specific points in the original memory. REPLAY II faithfully reproduces the lower 48K of memory in a fast load format. The upper 16K can also be copied for a 64K copy. Standard DOS 3.3 files are created automatically for storage on floppy or hard disks. A RAM card is needed for this.

REPLAY II is fully documented in a 60 page manual. Utility programs supplied with the REPLAY II card include Program Analysis, Comparisons, Packing and Compression. A language card is not needed to run packed program copies.

Because most programs are written in Assembly language, the user should be familiar with Assembly in order to fully utilize the advanced Analysis and Packing programs. Users can now freeze a binary program and perform a transparent step or trace while continuous disassembly is shown. View text or hires during trace.

REPLAY II can automatically move protected APPLESOFT programs to a standard DOS 3.3 disk for listing or modification.

Now game players can save a game at any level and QUICKLY restart with the REPLAY II card. Users can freeze games, change variables to obtain unlimited ships or power, etc., then restart the program. Saving high scores is easy!

Minimum requirements are an APPLE II and a single disk drive.

A Call —A.P.P.L.E. TECHNOTE:

IGNORING ?EXTRA IGNORED

Ari/Dov Freund

SCOTT Zimmerman admitted it bravely; I admit it too. He killed [Ctrl–C] (see: [Ctrl–C] KILLER by Scott Zimmerman, *Call –A.P.P.L.E.* December 1982), and I killed "?EXTRA IGNORED." The murder occurred on a common background, [Ctrl–C] was killing him and ?EXTRA IGNORED was killing a physicist I know.

During one of my visits to a nearby university, I met a physicist who was using his Apple to control his experiments. His problem was that while INPUTting numeric and alphanumeric strings, erroneously typing a comma (",") or a colon (":") would result in the ?EXTRA IGNORED message. Since ?EXTRA IGNORED errors are not detected by the ON-ERR command, entry of wrong data (only half of the string) results. The problem with numeric strings was the worst. Instead of hitting the period (".") key in entering floating point parameters, he would occasionally hit the comma which is right next to it, and the values entered came out wrong.

Another problem was that ?EXTRA IGNORED destroyed the entire screen format, which was organized in such a way that the cursor could be moved back up the screen to change data already INPUTed. This, of course, assumed the original line positions, which, however, were destroyed by ?EXTRA IGNORED. Worst of all there is absolutely no way in Applesoft to head these errors off.

If you have a similar problem, good news!: IGNORING ?EXTRA IGNORED has come to your rescue.

IGNORING ?EXTRA IGNORED is a small (44 bytes long) machine-language routine which intercepts the input coming in from the keyboard. If it finds either a comma, a colon, or a [Ctrl–C], it rings the bell to inform you of illegal input, cancels the character, and waits for you to input another one. To get IGNORING ?EXTRA IGNORED working, all you have to do is BRUN it from anywhere in memory. No need to POKE in anything at all, IGNORING ?EXTRA IGNORED will take care of itself. If you typed it into $300 as shown in listing 1, all you need do is CALL 768. Of course, if you want multiple INPUTs on a single line, you have to first disconnect IGNORING ?EXTRA IGNORED. To disconnect, you can either press RESET, or execute the IN#0 command. Note that IN#0 should be the DOS command, so if issued in a program it should have the form of PRINT CHR$(4) "IN#0".

```
:ASM
1                ORG   $300
2       *
        ********************************
4       *                              *
5       *   IGNORING  EXTRA IGNORED    *
6       *                              *
7       *    by Ari and Dov Freund     *
8       *                              *
9       * Call -A.P.P.L.E. :Sep. 1983  *
10      *                              *
11      *                              *
12      * This routine eliminates the  *
13      * "EXTRA IGNORED" message by   *
14      * intercepting commas and      *
15      * colons.  it rings the bell   *
16      * and discards them.  it also  *
17      * discards Ctrl-C.             *
18      *                              *
19      * The routine is BRUNable      *
20      * from anywhere in memory.     *
21      * to  connect -                *
22      * BRUN or CALL [start addr]    *
23      * to disconnect -              *
24      * use IN#0 (from keybaord)     *
25      * use PRINT CHR$(4)"IN#0"      *
26      * (from program)               *
27      * or simply press "reset".     *
28      *                              *
        ********************************
30      *
31      STACK     EQU   $100
32      DOSVEC    EQU   $3EA
33      RDKEY     EQU   $FD0C
34      KEYIN     EQU   $FD1B
35      BELL      EQU   $FF3A
36      WHERAMI   EQU   $FF58
37      KSWL      EQU   $38
```

How IGNORING ?EXTRA IGNORED Works — A Word About Input, KSW, and DOS

If you are already familiar with the above, you may skip on to *Final Comments on IGNORING ?EXTRA IGNORED.*

I am glad to see you are still with me. We'll start with INPUT AND KSW and then proceed on to DOS.

When Applesoft finds the INPUT command, among other things, it also jumps to the monitor's input routine GETLN1 ($FD6F). This gets a string from the keyboard and stores it in the input-buffer ($200–$2FF). GETLN1 is designed in a way that lets the Apple get input from all sorts of input devices, not only the keyboard. The way this is achieved when GETLN1 wants input, it calls RDCHAR ($FD35) which, in turn, calls RDKEY ($FD0C).

IGNORING ?EXTRA IGNORED

RDKEY in its turn does an indirect JMP to a routine which actually gets the input. This indirect jump goes through KSW. KSW stands for Keyboard input SWitch and is a two-byte address ($38-$39) which points to the final input routine, usually KEYIN ($FD1B). But, the nice thing here is that KSW can also be changed to point to any other routine such as one written by you, the user.

So far, so good, but now DOS enters from off stage. DOS always hooks itself to KSW and takes over all input. DOS does this so that when READing or EXECuting a textfile it knows to take the input from the disk, rather than from the keyboard. When not READing or EXECuting, DOS goes back and gets its input from the keyboard by calling KEYIN. However, it doesn't do this directly, but goes through KSW!!

This raises the question: How can DOS call KEYIN through KSW if KSW is already pointing to DOS? The answer is that upon entry from KSW, DOS temporarily places the final input routine address, usually KEYIN, in KSW, and upon returning to RDKEY (where DOS first came in), DOS rehooks itself. Of course DOS must meanwhile store KEYIN's (actually the current input routine) address somewhere. This it does at $AA55-$AA56. Therefore, all we have to do in order to get hold of the input before DOS grabs it, is to put our routine's address in these locations. When fiddling with DOS it is always wise to use DOS's page 3 vectors ($3D0-$3FF), so we place our address in KSW (lines 43-51), and then call $3EA (line 52) which completes the job.

Once called by DOS, our routine calls KEYIN (line 58) to get the input, which comes back in the A register. Next it checks for an illegal character (comma, colon, [Ctrl-C]) (lines 59-64). If it doesn't find one, it simply does a RTS, leaving the character in the A register for DOS to handle next. If an illegal character was entered, we ring the bell by calling $FF3A (line 56) and get a legal character by calling RDKEY (line 57). The reason we call RDKEY instead of KEYIN is that RDKEY first restores the cursor (which vanishes when a key is pressed) and then calls KEYIN through KSW, thereby cycling back to our routine.

Note that the address of KEYIN ends up in $AA55-$AA56 after RESET is pressed, or the IN#0 command is executed. This is why you can disconnect IGNORING ?EXTRA IGNORED by either of them.

Final Comments on IGNORING ?EXTRA IGNORED (Don't ignore these!)

Note that if you are having your INPUT coming in from a remote device (as a TELETYPE), you probably will not be able to use IGNORING ?EXTRA IGNORED since your driver will most likely disconnect it.

Another point to remember is that IGNORING ?EXTRA IGNORED might be overwritten by other programs, and if still hooked to KSW would cause the system to crash upon input request. Therefore, it's smart to disconnect it as soon as you're done with it.

If you are using a cassette instead of a disk simply replace line:

```
                        38  KSWH     EQU  $39
                        39  CTRLC    EQU  $83
                        40  *
                        41  * Initialization
                        42  *
0300: 20 58 FF          43           JSR  WHERAMI     ;Get address of program
0303: BA                44  TRANSFR TSX              ;From stack
0304: B5 FF             45           LDA  STACK-1,X
0306: 18                46           CLC
0307: 69 19             47           ADC  #CHEKCHR-TRANSFR+1
0309: 85 38             48           STA  KSWL        ;Take it out and store it
030B: A9 00             49           LDA  #$0
030D: 7D 00 01          50           ADC  STACK,X
0310: 85 39             51           STA  KSWH        ;In keyboard switch
0312: 4C EA 03          52           JMP  DOSVEC      ;Let DOS know
                        53  *
                        54  * Keyboard intercept routine
                        55  *
0315: 20 3A FF          56  INVALID JSR  BELL        ;Ring bell when invalid char found
0318: 4C 0C FD          57           JMP  RDKEY       ;And go get a new one instead
031B: 20 1B FD          58  CHEKCHR JSR  KEYIN       ;Input one char
031E: C9 AC             59           CMP  #","        ;Is it a comma ?
0320: F0 F3             60           BEQ  INVALID
0322: C9 BA             61           CMP  #":"        ;,or a colon ?
0324: F0 EF             62           BEQ  INVALID
0326: C9 83             63           CMP  #CTRLC      ;,or Ctrl-C ?
0328: F0 EB             64           BEQ  INVALID     ;Yes go handle it
032A: 60                65           RTS              ;All okay

--End assembly--

43 bytes

Errors: 0
```

```
52 JMP DOSVEC
```

with:

```
52 RTS.
```

Connecting IGNORING ?EXTRA IGNORED will be as previously described. To disconnect IGNORING ?EXTRA IGNORED simply hit RESET or type IN#0. Note that now the command is the Applesoft one, that is you use IN#0 rather than PRINT CHR$(4)"IN#0".

IGNORING ?EXTRA IGNORED can be placed anywhere in memory. However, in order to hook itself to KSW, it must first know where you put it. To do this, IGNORING ?EXTRA IGNORED uses the same method used by the ROM of the disk controller. It JSRs (line 43) to a location in the monitor ($FF58) which contains a RTS. (Any RTS would do.) After this RTS is executed, the address of IGNORING ?EXTRA IGNORED is now found in memory page 1 ($100–$1FF) in the two bytes below the location pointed to by the stack pointer. (This is true for any return address after a RTS.) Then IGNORING ?EXTRA IGNORED uses the TSX (line 44) instruction, gets its address out of page one, (lines 45–51) and finally stores it in KSW.

IGNORING ?EXTRA IGNORED is compatible with any size memory without modification.

IGNORING ?EXTRA IGNORED is built in a way that allows easy expansion. If you want more characters to be illegal, all you have to do is to insert before the RTS in line 65 the following lines for each new character: (for example, to exclude ".", whose ASCII code is $AE)

```
32B:C9 AE   65 CMP #"."
32D:F0 E7   66 BEQ INVALID
32F:60      67 RTS
```

If you want a large number of characters to be illegal, it might be a good idea to test for validity with a loop checking a data table. You could also eliminate all the letters, for example, by checking to see if the ASCII code of the input is in the range $C1–$DA. I'm sure you'll be able to develop many interesting variations to fit your own needs.

I hope you find this small utility useful and handy.

GOT A FUNNY DISK?

... WANT TO KNOW MORE ABOUT IT?

Then you need the CIA (Confidential Information Advisors)

CAN YOU . . .

* **edit normal or protected disks?**
* **quickly find and recover any intact file, however badly the disk is corrupted?**
* **list programs directly from any disk - protected or not?**
* **examine textfiles directly from any disk - protected or not?**
* **analyse the formatting of normal or protected disks?**
* **decrypt commercial software - or encrypt your own?**
* **rapidly auto-search normal or protected disks for anything you like?**
* **understand & use the latest copy protection methods?**
* **use your Apple™ as a powerful document retrieval system?**
* **make use of an exhaustive knowledge of disk lore?**

YOU CAN NOW — with a little help from these 5 sophisticated disk utilities:

TRICKY DICK examines, records, deletes, and edits. It can: (1) read individual sectors from normal and most protected disks, (2) list their contents in BASIC, assembler, ASCII, or hex, (3) edit them; (4) write them back to the disk. Tricky Dick cunningly bypasses most protection systems, allowing you to work on disks with nonstandard formatting, half-tracks, and altered DOS marks. It is also a chief executive program that directs the following undercover agents:

THE LINGUIST reads in a trackful of raw data for your scrutiny, translates all the address information, and allows you to inspect the track's formatting. It also translates all 3 types of DOS encoding (6 & 2, 5 & 3, 4 & 4), and works with Tricky Dick to list and examine programs or textfiles on any protected disk. You can use The Linguist to recover valuable files from blown disks, improve your programming skills by studying commercial software, and analyse standard or altered formatting.

THE TRACER rapidly searches normal and most protected disks for up to six strings of your choice simultaneously (specified in ASCII or hex). The Tracer also verifies disk formatting, and sniffs out all hidden catalog or VTOC sectors. When it finds something, it transfers control to Tricky Dick and puts the cursor over the object of your search. A few further keystrokes allow you to make any necessary changes and write the sector back to the disk.

THE CODE BREAKER keeps your programs and textfiles from prying eyes by enabling you to translate them into a "secret code" during disk storage. This utility also deciphers encrypted commercial programmes, allowing you to use Tricky Dick to read, list, and edit software never before accessible to any disk utility.

THE TRACKER closely shadows the disk drive arm, carefully recording all its movements and operations. The Tracker's job is to display, on either your screen or printer, a list of every track and sector accessed during a LOAD, RUN, SAVE, or any other DOS operation. This utility also tells you exactly where a read or write occurred during any disk access. Use The Tracker's services to locate the precise trouble spots on a clobbered disk, to determine sector skew patterns, to discover the location of hidden "nibble-count" tracks on protected disks, and to learn much more about how DOS works. You'll be surprised to see just exactly where the disk arm really does go!

What's more, you get permanent access to:

THE CIA FILES, a 50,000+ word book designed to turn you into a disk expert. In addition to complete instructions for the 5 CIA utilities, the book contains an easy-to-follow hand-holding tutorial (written in plain English!) on all aspects of the Apple disk. Using the CIA utilities as your personal guides, you progress step-by-step to total disk mastery. You'll acquire a wealth of skills and information relating to disk repair and file recovery, DOS patches, copy protection, disk formatting, program encryption, and other vital topics. Much of the material has never before appeared in print.

All programs are UNPROTECTED, and hence can be copied, listed, and modified at will. (special patches are described in the manual). They require one drive, DOS 3.3, and 48K of RAM.

— TO GET THE CIA ON THE TRAIL OF YOUR DISKS, —
SEND $65.00 Money Order, (checks allow time to clear), to:
GOLDEN-DELICIOUS SOFTWARE LTD.
350 FIFTH AVENUE, SUITE 3308, NEW YORK, N.Y. 10118
No Credit Cards Please

Circle 1 on Reader Service Card

ADVERTISEMENT

Educational News

Orange Plus Educational News located at 23801 Calabasas Road/Suite 2050/Calabasas, CA 91302/(213) 999-5210

Legal Apple[Type] Compatible

New Computer runs CP/M & Apple[Type] Software

CALABASAS — Now instructors can teach Apple-type programs such as LOGO in one class and professional-style word processing in another without any hardware changes. Dual microprocessors (Z80-A & 6502, two computers in one case), with the radical new Orange Plus development the "EuroROM", allows the machine to read/write/work with Apple-type software as well as CP/M programs and access either CPU via the keyboard.

The ***"ORANGE+TWO"***™ is a brand new direction in the evolution of the personal computer. ORANGEFORTH-83, a derivative of Fig-FORTH — a readily available public domain language, is resident in the ROM*. Also included is CP/M 3.0, Digital Research's latest CP/M version. There is also a built-in disk drive controller for two Apple-type drives, a cassette interface, joystick port, color graphics and ASCII keyboard with numeric keypad. For expert word processing, the keyboard features upper and lower case (lower case characters are true descenders) with auto repeat.

This new breed of computer is a breakthrough for educators and school systems throughout the world. The ability to run both Apple-type and CP/M software on the same machine relieves financially-pressed educators from expensive equipment burdens, allowing them to spend more money where it counts...**on the teachers.**

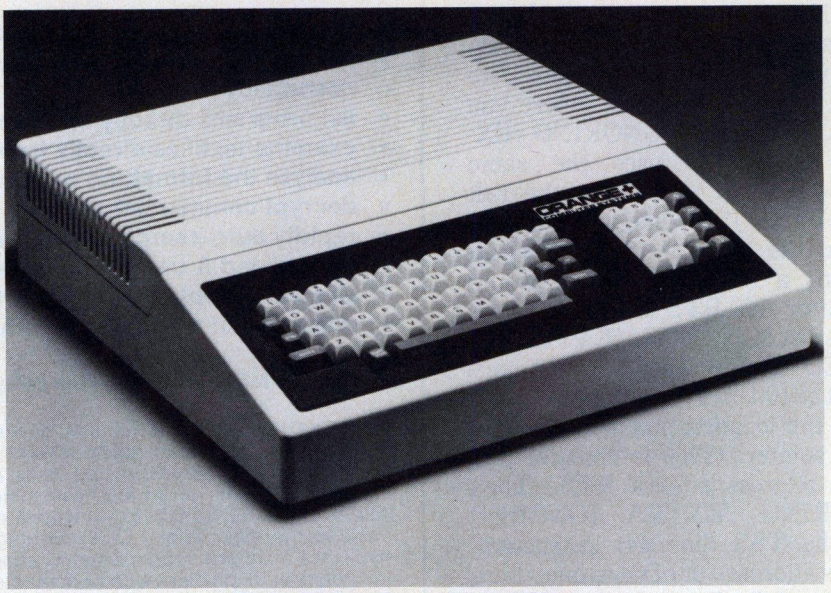
*Available on disk at extrta cost.

THE LEASING ALTERNATIVE

CALABASAS — Through select leasing companies, Orange Plus Computer Systems will offer the following lease programs to qualified corporations and educational institutions: An ***"ORANGE+TWO"***™ computer with a green monitor and one disk drive for only $59.55 per month...36 month closed end lease. Also included is a full maintenance program and all revisions and upgrades that may be available during the term of the lease. Subject to credit approval. Call for additional information.

DISTRIBUTORS: CALL (213) 999-5210

CALABASAS — Due to the tremendous amount of individual inquiries and dealer applications, qualified stocking distributors are needed. Select areas available!



MORE FOR LESS

The ***"ORANGE+TWO"***™ 64K computer includes these standard features:

Built-in CP/M 2.2 compatibility	**no charge**
Digital Research's CP/M 3.0	**no charge**
Built-in disk drive controller for two Apple-type drives	**no charge**
Numeric keypad with separate return key	**no charge**
110/220 volt switch selectable power supply, 50-60 Hz	**no charge**
Full function ASCII keyboard with auto repeat	**no charge**
Cassette interface	**no charge**
Fully grounded metal base plate	**no charge**
Adjustable audio volume control	**no charge**
ORANGEFORTH-83, Z80-A FORTH language, resident in ROM	**no charge**
Z80-A CPU, a second computer	**no charge**
6 slot double sided logic board	**no charge**
Bank switchable RAM, fully socketed, expandable to 256K	**no charge**
Programmable 2764 EPROMs	**no charge**
TOTAL: *"ORANGE+TWO"*™	**$1095.00**

Compare these features included on the ***"ORANGE+TWO"***™ *with the competition's over $2000 for equivalent product.*

MORE PRODUCTS FROM ORANGE PLUS COMPUTER SYSTEMS TO ENHANCE YOUR *"ORANGE+TWO"*™, APPLE II/II PLUS, OR FRANKLIN ACE 1000

Disk-based ORANGEFORTH-83 language with full documentation	**$99.95**
Digital Research's™ CBASIC with disk and documentation	**$49.95**
Orange Plus KoalaPad™ Touch Tablet w/Micro Illustrator™	**$124.95**
Orange Plus Joystick (works on "ORANGE+TWO", Apple II, II+, & //e)	**$29.95**
Orange Plus self-centering joystick	**$39.95**
10MB (Formatted) 5¼" half-height Winchester Hard Disk Drive	**$1495.00**
Controller & Interface for Hard Disk**	**$395.00**

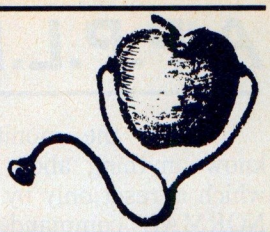
Includes everything necessary to be installed in* ***"Orange+Two"*, Apple or Franklin Computer*

The above are suggested retail prices. Prices may vary from state to state.

Collins International Trading Corporation,
23801 Calabasas Road, Suite 2050, Calabasas, CA. 91302
(213) 999-5210

Circle 31 on Reader Service Card

A.P.P.L.E. DOCTOR



Bruce Field

Question:

I want to do some work in Esperanto, which requires superscripts (umlaut, accents, etc.) above the characters. I think the Apple Dot Matrix printer will do this, but will I be able to see it on the screen? A British Esperantist offers the necessary ROM to do this for the EPSON MX80. Would this do me any good? Also, I could use a German alphabet right now for occasional words and wonder if it's worth the trouble.

Answer:

The May 1983 issue of Apple Assembly Line contains an ad for a Font Downloader & Character Editor for the Apple Dot Matrix Printer or C.Itoh 8510AP printer. This program ($39.95 from RAK-WARE, 41 Ralph Road, West Orange, NJ 07052) allows you to generate a character font containing any special characters or symbols you need. This new font can be quickly downloaded to the printer before using your word processor. Then during printing the word processor should output ESC ' to turn on the custom font and ESC $ to return to the standard font. Unfortunately you won't be able to display these characters on the screen using this method.

There are several foreign language characters normally available with the Apple Dot Matrix Printer, however the documentation is not clear on how to access these additional characters. The values for the sets are:

0 - English (US)	1 - Italian
3 - English (UK)	4 - German
5 - Swedish	6 - French
7 - Spanish	

To turn on the alternate character set from Applesoft, assuming you are already in English (US), all you have to do is PRINT CHR$(27)"Z"CHR$(0);CHR$(0); CHR$(27)"D"CHR$(N);CHR$(0) where N is the number of the set above. To return to English (US): PRINT CHR$(27)"Z"CHR$(N);CHR$(0);CHR$(27)"D"CHR$(0);CHR$(0) where N is the number of the set you were previously in and are now turning off. To output these control characters from a word processor may be a little difficult, you will have to consult your word processor manual to find out how to imbed control characters.

Question:

We have an Apple][Plus that has a hardware problem. When we first turn on the Apple on cold mornings we get vertical bars on the screen instead of any text. Programs seem to work properly except we can't see the text. Also the graphics screen seems to work okay, and when the Apple warms up the vertical bars disappear and the text will appear. We have a service contract with a local computer store but when the repairman came out the weather was warmer and the problem disappeared. He ran a diagnostic program and everything checked out okay. Can you help?

Answer:

Since the graphics screen seems to be okay to problem is almost certainly with the character generator ROM. This is a 24-pin chip located on the motherboard, underneath the keyboard. Specifically, it is at location A5. For early Apple][+'s it is a 2513 ROM, and for later models (rev. 7 and later) it is a 2316 ROM.

The diagnostic disk can only check components and peripherals connected to the Apple that are accessible to the CPU chip. For example, the Apple can send a command to the printer to print a certain character, but it has no way of looking at the character on the paper to see if the ribbon is worn out or not. Similarly, it can put a character in memory so that it should be displayed on the screen, but it has no way of knowing if it is actually displayed properly. The TV or monitor could be turned off and the Apple wouldn't know it. When using a diagnostic test for printing to the screen, it is up to the user to tell the computer whether or not everything is okay.

Question:

The other day I ran across what seems to be a bug in Applesoft or maybe DOS. If the Applesoft text output mode is INVERSE or FLASHING, then file operations behave in a most peculiar way. Mainly they don't work. If FLASH is invoked in a program and then a text file is OPENed and READ before NORMAL is called, DOS will open a file with an $E0 prefixed to the intended name. DOS then of course tries to read from a newly opened file that is empty and goes into error. Also a one sector text file is created with what appears to be the intended name having a space as the first character. The file cannot be deleted from DOS unless it's apparent name is concatenated to an $E0.

Answer:

(Thanks to Val Golding for the answer to this one.) The problem results from a number of circumstances none of which directly relate to Applesoft. To output flashing characters, Integer Basic and the monitor store the value 127 in location 50 (POKE 50,127). It is cleared with POKE 50,255.

Owing to a bug in the monitor, only ASCII characters with a value less than 160 can be displayed in FLASHING. The space, figures and special characters are displayed in inverse only.

The FLASH routine in Applesoft overcomes this problem by placing a value of 64 in location 243 when FLASH is called. This location is called ORMASK and its function is to add 64 to the ASCII value of those characters which do not flash properly.

A.P.P.L.E. DOCTOR

Neither the monitor nor DOS know anything about location 243, which is reset only by the Applesoft NORMAL command. When DOS receives a request for a file access, it stores 255 in location 50, which theoretically restores the characters to normal. However, since ORMASK is not restored to 0, those characters with a value of 160 and greater still have 64 added to them, thus $A0 [space] becomes $E0, etc.

Question:

After carefully typing in a long program from a magazine and saving it to disk, I find that I cannot seem to load it back into memory. The disk spins for a while then beeps and prints I/O ERROR. I remember reading somewhere that it is possible to do some pokes so the disk will read even with an error. Do you know what these pokes are?

Answer:

When DOS writes information from memory to the disk, it does so a sector at a time, where a sector is a group of 256 bytes of data. In addition it also writes a checksum byte that is a combination of the 256 data bytes just written. When the data is read back a new checksum is computed and compared to the original checksum written on the disk. If the two do not agree, it means that there is an error in the data. When this occurs DOS will try to re-read the sector up to 48 times before giving up and stopping with an I/O ERROR. We will modify DOS to re-read up to 48 times and if there is still an error we will ignore it and return with whatever data we have read. The following two POKEs will do this:

POKE 48590,48 : POKE 48591,29.

You only want to do this for disks that have I/O errors and you should carefully check the data for errors once it is read in. After you have done this, save the data or program on a new diskette and re-boot a new copy of DOS to remove the POKEs.

Question:

A recent attempt to read a random access file, which I knew had 47 records in it, stopped at record 29 with an END OF DATA error. Out came the back up disk with the same result.

By trying to read individual records, it was determined that records 29, 30, and 31 were the only bad ones, and with a record length of 80, it seemed likely only one sector was involved.

Using Disk Zap, I read the sector that should have contained the missing records, and I was stunned to find the recently revised menu from another program. The gentleman I consult at my local computer store said that for some reason the VTOC had mistakenly marked the sector as being free, and suggested that I run RECONSTRUCT VTOC. The RECONSTRUCT VTOC program came up with an incredible 46 errors corrected. How can this be when I apparently had only the one problem, and what kind of error was it talking about? Would it be a good idea to routinely run RECONSTRUCT VTOC on important disks with frequently revised data? What could I have done that cause this problem?

Answer:

I'll answer your questions in reverse order. Opening the disk drive door or pressing RESET while the computer is writing to the disk could give you problems. It is also possible that while debugging a program, it somehow wrote into memory incorrectly somewhat scrambling DOS, then when you saved the program it was saved to the wrong area of the disk. It is a good idea to use "disposable" diskettes when you are writing and debugging programs and keep your good records on separate disks. It is also possible that you did nothing to cause this problem, an electrical spike on the power line or a momentary problem with the disk controller card-edge connector could also be at fault.

I am not familiar with RECONSTRUCT VTOC although I can make a guess at what it does. It probably goes through the diskette directory making sure that the pointers to the data sectors are in order, and the sectors being used are marked as used in the VTOC. Although this will catch some errors, it would still be possible to have incorrect or scrambled data and pass RECONSTRUCT VTOC without any errors. The best solution is to keep backups of your current data as well as a copy of your data before the current update.

As far as having 46 errors, the data in sectors marked as free remains intact until new information is saved into that sector. You can read it as many times as you want without any problem until you save new information to the diskette, then some of the sectors *may be overwritten.*

I ran across an interesting phenomena the other day I'd like to pass along. Applesoft programmers know the INPUT command stumbles over commas and colons, returning an exasperating EXTRA IGNORED message when trying to input strings containing these characters. As a result, a spate of "input anything" routines have sprung up in the literature. (See for example *Call -A.P.P.L.E. in Depth No. 1:* All About Applesoft, p.97, *Call -A.P.P.L.E.* March-April 1981, p.54, *Call -A.P.P.L.E.* January 1983, p.43, and Nibble Vol. 2, No. 2, p.57) All of these require loading or poking a machine language routine into memory, which is a minor pain. Some also perform string manipulation, which quickly loads memory with extraneous strings, causing premature garbage collection. There is a much simpler way for a lot of applications.

To input a string containing commas or colons using the Applesoft INPUT command, preface the string with the quotation mark ("). For example if you want to input the following string:

John Doe, Anytown

using the INPUT A$ command, type:

"John Doe, Anytown[return]

The entire string INCLUDING the comma but EXCLUDING the " is placed in A$.

This also works for DOS text files! When originally printing the string to the disk include a " as the first character of the string. Unfortunately you can't print the " directly but instead have to use the CHR$ function. Like this:

```
PRINT CHR$(34);A$
```

When you read the string back, all the commas and colons come back perfectly. How come this works? It is all very carefully explained in the Applesoft Reference Manual (which people rarely read, me included). Quoting from page 66.

"...a response assigned to a string variable must be a single string or literal, not a string expression. Spaces preceding the first character are ignored. If the response is a string, then a quotation mark anywhere within the string will cause a ?RE-ENTER message. However, within a string, all characters except the quotation mark, [Ctrl-X] and [Ctrl-M] are accepted as characters for the string. This includes the colon and the comma. Spaces following the final quotation mark are ignored."

Well, well, if colons and commas *are* explicitly accepted how come John Doe, Anytown gives an EXTRA IGNORED message? The answer lies in the definition of a string and a literal. Back to the manual, pp. 145, 149 for the following definitions.

```
string
:= "[{character}]" or
:= "[{character}] <return>

literal
:= [{character}]
```

Ah, Ha! Strings are groups of characters enclosed in quotes, or prefaced by a quote and terminated by a carriage return. A literal on the other hand is just a group of characters with no quotes. Thus John Doe, Anytown is a literal, and again quoting from the manual (p. 66).

"If the response is a literal, then quotation marks are accepted as characters in any part of the literal except the first non-space character. Spaces following the last character are accepted as part of the literal. However, the comma and the colon (and [Ctrl-X] and [Ctrl-M]) are not accepted as characters in the literal."

The first sentence is a little confusing. You can put quotation marks in a literal and they will be accepted as characters, but you can't put quotation marks in a string except to denote the beginning or end of the string. If you put a quotation mark as the first character of a literal, by definition it is a string and no longer a literal. So, you can imbed quotation marks in a literal but not commas or colons. Boy, talk about picky computers!

Since I'm now totally confused, let me summarize all this. If you respond to a string INPUT where the first non-blank character is a ", all characters EXCEPT [Ctrl-X], [Ctrl-M] (a return), or " may be included in the response. If the first non-blank character is not a " commas, colons, [Ctrl-X], and [Ctrl-M] may *not* be included in the response. This works for both a keyboard response and a DOS text file.

$249.95 64K option

$329.95 128K option

VIEWMAX-80e

MOST ADVANCED 80-COLUMN EXTENDED VIDEO CARD FOR APPLE IIe

Expands Apple IIe MEMORY SPACE UP TO **192K!**

Plus adds 80-COLUMN VIDEO function usable as FAST-ACCESS DISK emulator

- 64K Configuration User-Upgradable to 128K
- Plugs into the Auxiliary Slot in Apple IIe
- Fully Compatible w/all Apple IIe 80-column Application Software including Magicalc & Magic Window II
- Bank Indicator Light for each 64K Segment

FULL TWO-YEAR HASSLE FREE WARRANTY

Contact us for our very healthy dealer/distributor discounts.

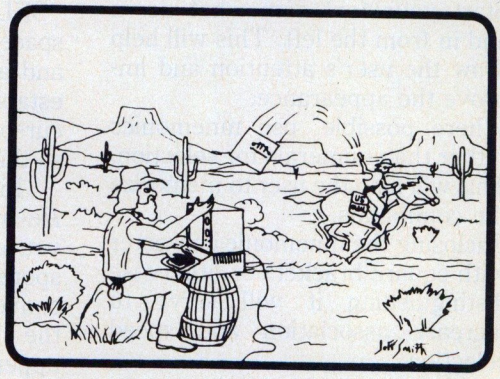
micromax™

INNOVATORS IN MICRO-COMPUTER TECHNOLOGY

6868 Nancy Ridge Drive San Diego, CA 92121 **619-457-3131**

Circle 22 on Reader Service Card

TELEPHONE SOFTWARE CONNECTION™ INC



Mail delivery also available. Call (213) 516-9430 (voice) or write for our FREE catalog.

Software delivered via your MODEM.
Call us now for some FREE samples:

SHIFT I N #3 #3 RETURN
CTRL A CTRL F CTRL A CTRL Q 1-213-516-9432

POST OFFICE BOX 6548 - TORRANCE, CALIFORNIA 90504

© 1983 Telephone Software Connection, Inc.

Circle 47 on Reader Service Card

A Call —A.P.P.L.E. TUTORIAL:

PREPARING A MENU

Val J. Golding

A properly designed menu ideally should include features to make things easy for the end user. Many of the applications programs we have written for internal use have been lacking in this area. More correctly, we made things too easy, meaning, for example, you could enter some incorrect data and have no way to fix it, or you could perhaps (fatally) make a wrong menu selection.

Why was this so? We're a programmer. Programmers are notorious for writing programs that work just great for themselves. We had this pointed out to us in no uncertain terms by our staff. We have turned over a new leaf! We promise on a stack of 6502's to bear this in mind when writing future programs.

What we would like to do here is to offer a check list of do's and don'ts in menu writing, and also show you a way to use logical variables in menu selection.

- Get all the junk off the screen. A neat, uncluttered screen will be easier for the user to read.
- Start your text somewhat down and in from the left. This will help draw the user's attention and improve the appearance.
- Where possible, use mnemonics rather than numerals for selection. This will help the user to memorize the commands.
- Enclosing the mnemonic character with square brackets or otherwise distinguishing it will serve to increase association and speed selection.
- Don't use flashing characters. They distract and should be reserved for emergency situations.
- Inverse mode should be used with discretion, for the same reason. We prefer to use it only for the block that reads "select one".
- Try to cover all bets. Maybe you won't type the wrong key, but figure you've got to be better informed about the program than the user, so use adequate error trapping.
- Don't leave the user wondering what's happening while a file is loading or an obscure, esoteric algorithm is slowly executing; print a "please stand by" type of message.
- Start your menu on an even thousand line number. If the user exits the program by mistake, he/she may be able to type "GOTO x000" to return to the program with data intact.

The program listed here is a menu excerpted from one of our own application programs, without modification.

Line 9010 clears all prior variables upon return to the menu from another section of the program. This may or may not be desireable, since all variables will have to be re-declared. This will be determined by your specific use.

Lines 9020–9040 create your screen format. The two POKEs in 9030 simulate HTABs in each of the menu item lines. They are cancelled by the TEXT in 9200. Lines 9050 to 9090 are the actual menu items. Line 9200 prints the inverse text "SELECT ONE". Note the inverse spaces, for appearance's sake, before and after the text. The variable VT establishes the current position of the cursor, prior to the TEXT command, so that we can restore it in line 9210.

Line 9300 was added for Apple *IIe* owners or GPLE users. In case you enter a lower case letter in response to a menu choice, this will convert it back to upper case, since the choice routine is looking only for upper case characters.

Line 9310 uses logical variables to determine which key was pressed, and this little gem frees you from the need to use numbered choices to accommodate the ON GOTO expression in line 9320. Each of the expressions within the parentheses will set the variable MENU to 1 if true, to 0 if false. Therefore, if CHOICE$ is equal to "Q," a 1 will be returned and the other logical expressions will be zero. Take a pencil and paper now and do the simple arithmetic in this algorithm and you will see not only how it works, but why you never again need to rely on numerals for menu choices.

Line 9330 is a catch-all, the error trap in case a wrong key is typed. It re-draws the menu screen, then forces the user to re-select.

That's it. So simple, if you will just place yourself in the user's chair and try to anticipate all of the goofs he or she may make. Apply this technique to each of your program modules and you'll never go wrong.

```
]LIST

9000  REM

MAIN MENU

9010  CLEAR :D$ =  CHR$
      (4)
9020  TEXT : HOME :
      VTAB 4:
9030  POKE 32,6: POKE
      33,34
9040  PRINT
9050  PRINT "[D] Delete
      a file "
9060  PRINT "[I] Initialize
      a file "
9070  PRINT "[Q] Quit
      program "
9080  PRINT "[S] Sort
      a file "
9090  PRINT "[U] Update
      a file "
9200  PRINT : PRINT
      "     ";: INVERSE
      : PRINT " SELECT
      ONE ";: NORMAL
      : PRINT " ";:
      GET CHOICE$: PRINT
      CHOICE$:VT =
      PEEK (37): TEXT

9210  VTAB VT
9300  IF  ASC (CHOICE$)
      ) 95 THEN CHOICE$
      =  CHR$ ( ASC
      (CHOICE$) - 32)
9310 MENU = (CHOICE$
      = "D") + 2 * (CHOICE$
      = "I") + 3 * (CHOICE$
      = "Q") + 4 * (CHOICE$
      = "S") + 5 * (CHOICE$
      = "U")
9320  ON MENU GOTO 1000,2000
      ,3000,4000,5000
9330  GOTO 9000: REM
```

CP/MAGIC:

A SOFTCARD LST DRIVER

Gerald Cahill

Introduction

MICROSOFT'S Softcard is delivered ready to drive printers which use the more common serial and parallel interface cards. If your I/O card is not on their list, you must add a user I/O driver such as LSTDR to the I/O Configuration Block on your system disk.

LSTDR will drive any printer which was activated with the "PR#1" command under Apple's DOS. Some of you may have ROMs on your printer controller card with "special features" burned into them which make complete compatibility impossible. If so, this driver can still be used until you can get a new ROM or learn to live with the minor differences.

This article is an overview of the driver's operation. A detailed description of how the code works is the assembly language listing itself. We will review how Apple DOS interfaces with your printer's ROM, how this driver fools your ROM into thinking DOS is still alive, what the possible incompatibilities may be, and finally, how to install this driver on to your system disk.

Remember Apple DOS

When you issue a PR#1 command under DOS, the address of the first location of your printer's ROM is stored in Apple Monitor locations $36 and $37 (CSWL and CSWH). When the first character is sent to your printer, the software in ROM will initialize itself and then set CSWL and CSWH to point to a different location in the ROM. DOS notices the pointer change and remembers it so that additional characters will not cause the ROM to initialize itself again. When you shut off your printer with a PR#0 DOS command, the pointer in $36 and $37 is reset to the address of the screen output routine in the Apple Monitor.

Fool Your ROM

The simplest way to use your ROM from the Softcard is to call it in the same manner that DOS does. When LSTDR is entered, the ASCII character to be printed is in the C register. Then the high order bit of the character is set to one since most ROMs expect that. The current value of the pointer in $36 and $37 is saved and a pointer to your ROM is stored there. The new pointer is also used to tell the Softcard where to start the 6502 executing. The character is passed to the 6502 in the A register (where your ROM expects it to be) and the 6502 is turned on. When your ROM does its final RTS, the Softcard is turned on. The possible new value of $36 is saved so that the next character printed will not cause your ROM to initialize itself again. The old pointer is restored to $36 and $37, and this driver returns to the caller. Sound simple, right? Well, watch out for the simple programs; this one took six evenings to complete.

Automatic Line Feed Problem

All printers used with the standard software supplied with the Apple must supply a line feed character following each carriage return character printed. A lot of the printers have this line feed supplied by the ROM driver.

```
                        ; SOFTCARD - DRIVER FOR LST DEVICE.

                        ; PRINTER DRIVER MUST BE IN ROM (256 BYTE) ON THE CONTROLLER CARD.
                        ; CARD MUST BE IN SLOT 1 (WAS ACTIVATED UNDER DOS BY PR#1).

                        ; SET CRLF TO 0 IF YOUR PRINT DRIVER DOES NOT AUTOMATICALLY FOLLOW
                        ; EACH CARRIAGE RETURN WITH A LINE FEED CHARACTER.
                        ; SET CRLF TO 1 IF YOUR DRIVER SUPPLIES THE LF AND YOU WANT THIS
                        ; ROUTINE TO SKIP THE LINE FEED WHICH FOLLOWS A CARRIAGE RETURN.

0001 =                  CRLF EQU 1

                        ; SET ECHO TO 0 IF YOUR PRINT DRIVER DOES NOT ECHO EACH
                        ; PRINTED CHARACTER TO THE SCREEN.
                        ; SET ECHO TO 1 IF YOUR DRIVER DOES ECHO TO THE SCREEN.

0001 =                  ECHO EQU 1

                        ; ONE LAST THING.
                        ; WHEN YOU DID A PR#1, DOS JUMPED TO $C100 AND THE DRIVER INITIALIZED.
                        ; THIS DRIVER JUMPS TO $C100 AFTER A COLD-BOOT.
                        ; SOME DRIVERS MUST BE INITIALIZED AFTER THE RESET KEY IS PUSHED.
                        ; THE SOFTCARD DOES A WARM-BOOT WHEN THE RESET KEY IS HIT (UGH).
                        ; SO, IF YOUR DRIVER MUST BE RE-INITIALIZED AFTER A RESET,
                        ; YOU MUST PLACE SOME CODE AT LOCATION LSTDR: TO SET THE VARIABLE
                        ; LASTL TO ZERO IF RESET WAS PUSHED.  SEE THE EXAMPLE BELOW.

                        ;       GERALD CAHILL . . .
                        ;       2069 SPICE STREET
                        ;       LANCASTER, CA.  93534
```

A SOFTCARD LST DRIVER

The Softcard (and most everyone else) use a carriage return character only to return the carriage and the line feed only to feed a line. Some word processors use the carriage return not followed by a line feed to implement bold face printing. Turn off the automatic line feed option in your printer if you can. If you can't, LSTDR will ignore the first line feed following each carriage return.

EECCHHOO

Most ROM drivers will echo each printed character to the screen. This driver can not stop the echo but it will nullify its effect. Because of the method the Softcard uses to write to the screen, a simple backspace and write new cursor sequence will not always leave the screen in the desired state. The way this driver operates is to set up the Apple Monitor to write the character being echoed to a nice safe screen position (I chose the second character in the top line to avoid any problems with automatic scroll or backspace). First the current second character and the screen pointers in Monitor are saved. Then the screen pointers are sent to the second character position. Now the character to be printed is passed to your ROM. When your ROM returns to this driver, the screen pointers and the second character are restored to their former values.

Reset Key Problem

When the Softcard system is first booted, it is a cold-boot. That is, all of memory is initialized. When the RESET key is pushed, the Softcard does a warm-boot (just like pushing a [Ctrl-C]). The point is the I/O routines are not reinitialized when RESET is pushed! Wonderful for the software, but when you hit RESET, a hardware signal is sent everywhere including to the boards in the I/O slots. Some printer controller boards must be reinitialized after a RESET. Somehow LSTDR must determine when a RESET has occurred and then set the pointer in $36 and $37 to $C100 so that your ROM will initialize itself again (like a PR#1 would do).

If you are lucky, your controller will not care about a RESET and you don't need to add any code to this driver. On the other hand, if a RESET zaps your controller, you must place some code at the start of LSTDR to force a pointer value of $C100 whenever RESET has been pushed. I have included some code as comment lines in the listing which does the job for my controller. The code simply assumes the RESET has been pushed if the printer does not become ready within a half second. That's enough time for the printer to finish handling the last character even if it was a carriage return.

If you insert your own code, be sure the length of LSTDR does not exceed 7F hex.

Installation Procedure

1) ED LSTDR.ASM (To create a file which contains the source code.)
2) ASM LSTDR (To create file LSTDR.HEX)
3) LOAD LSTDR (To create file LSTDR.COM)
4) MBASIC CONFIGIO.

```
F3DE =          Z$CPU EQU 0F3DEH   ;LOC OF SOFTCARD STORED HERE.
F3D0 =          A$VEC EQU 0F3D0H   ;ADDR OF 6502 SUB TO CALL GOES HERE.
F045 =          A$ACC EQU 0F045H   ;6502 A REGISTER GOES HERE.
F036 =          A$CSW EQU 0F036H   ;APPLE MONITOR'S CSWL.
FB5B =          A$TABV EQU 0FB5BH   ;MONITOR'S TABV ROUTINE.
F024 =          A$CH EQU 0F024H   ;MONITOR'S CH.
F025 =          A$CV EQU 0F025H   ;MONITOR'S CV.
F401 =          A$TUBE EQU 0F401H   ;LOCATION OF 2ND CHAR ON TOP LINE.
008D =          CR EQU 08DH   ;ASCII CR WITH HIGH BIT SET.
008A =          LF EQU 08AH   ;ASCII LF WITH HIGH BIT SET.

0100             ORG 100H   ;FORMAT REQUIRED BY CONFIGIO PROGRAM.
F200 =          ORIGIN EQU 0F200H   ;SLOT 1 - LST DEVICE CP/M AREA.
                OFFSET SET ORIGIN-LSTDR ;SET, NOT EQU.

0100 01          DB 1   ;MAKE 1 PATCH.
0101 00F2        DW ORIGIN   ;DESTINATION ADDRESS TO PUT LSTDR.
0103 7C00        DW LSTEND-LSTDR    ;LENGTH OF LSTDR. MUST BE LESS THAN 80H.
0105 01          DB 1   ;PATCH TYPE 1.
0106 0A          DB 10   ;LIST OUTPUT VECTOR #1.
0107 00F2        DW ORIGIN   ;ADDRESS OF USER CODE (AGAIN).

                ; CP/M ENTERS HERE WITH CHARACTER TO PRINT IN C REGISTER.

                LSTDR:

                ; EXAMPLE OF CODE TO DETECT AND RECOVER FROM A PUSH OF THE RESET KEY....
                ; IF RESET DOESNT ZAP YOUR ROM DRIVER, YOU DONT NEED ANY CODE HERE.
                ; I ASSUME RESET WAS PUSHED IF THE PRINTER IS NOT READY FOR TOO LONG.
0109 11FF3F      LXI D,03FFFH ;SET TIMEOUT COUNT.
010C 3A90E0     LOOP: LDA 0E090H   ;GET MY PRINTER'S STATUS.
010F E640        ANI 040H   ;CHECK THE NOT READY BIT.
0111 CA15F2      JZ NORST+OFFSET ;JUMP IF PRINTER READY (IF BIT IS 0).
0114 1B          DCX D   ;WAIT FOR TIMEOUT.
0115 7A          MOV A,D
```

Tell CONFIGIO whether or not you have a lower case adapter, and then run function 3 giving LSTDR as the name of the file to load. Then run function 4 to write the new I/O Configuration Block on your system disk. Finally type Q to get back to MBASIC.

While in MBASIC, try the LPRINT command. Your printer should come to life. Type SYSTEM to get out of MBASIC. When back under the operating system, type a [Ctrl-P] and see if a copy of everything sent to the screen is also printed. Another [Ctrl-P] will stop the printer.

Well, Did It Work?

I would like to hear about any problems you have with this driver and any suggestions you have which will help others to get their printers running.

```
0116 B3          ORA E
0117 C203F2      JNZ LOOP+OFFSET
011A AF          XRA A  ;TIMEOUT. SET LASTL=0
011B 3279F2      STA LASTL+OFFSET ;TO FORCE ENTRY AT $C100.
                 NORST:
                 ; END OF SPECIAL RESET RECOVERY CODE.....................................

011E 79          MOV A,C  ;GET ASCII CHARACTER IN (A).
011F F680        ORI 80H  ;SET HIGH BIT OF CHARACTER.
0121 4F          MOV C,A  ;SAVE CHARACTER IN (C).

                 IF CRLF
0122 FE8A        CPI LF   ;IS IT A LF.
0124 C22BF2      JNZ GO$ON+OFFSET    ;NO, JUMP TO GO$ON.
0127 3A7BF2      LDA LASTC+OFFSET    ;YES, CHECK TO SEE IF
012A FE8D        CPI CR   ;LAST CHAR SENT WAS A CR.
012C C22BF2      JNZ GO$ON+OFFSET    ;NO, JUMP TO GO$ON.
012F 79          MOV A,C  ;GOT CR,LF SO SKIP THE LF.
0130 327BF2      STA LASTC+OFFSET ;BUT REMEMBER THE LF.
0133 C9          RET    ;RETURN WITHOUT SENDING LF.
                 GO$ON:
                 ENDIF

                 IF ECHO
0134 3A24F0      LDA A$CH   ;SAVE CURRENT VALUE OF CH IN (D).
0137 57          MOV D,A
0138 3A25F0      LDA A$CV   ;SAVE CURRENT VALUE OF CV IN (E).
013B 5F          MOV E,A
013C AF          XRA A  ;SET A REG TO ZERO.
013D 215BFB      LXI H,A$TABV ;LET MONITOR SET CV STUFF TO LINE 0.
0140 CD6EF2      CALL APPLE+OFFSET
0143 3E01        MVI A,1  ;SET CH TO 1.
0145 3224F0      STA A$CH   ;(THE 2ND CHARACTER).
0148 3A01F4      LDA A$TUBE   ;SAVE THE CURRENT 2ND CHARACTER IN (B).
014B 47          MOV B,A
                 ENDIF

014C 2A36F0      LHLD A$CSW  ;SAVE CURRENT CSWH AND CSWL
014F E5          PUSH H  ;ON THE STACK.
0150 2A79F2      LHLD LASTL+OFFSET   ;GET LAST VALUE ROM SET CSW TO.
0153 2236F0      SHLD A$CSW  ;STORE POINTER TO ROM IN CSW.
0156 79          MOV A,C  ;GET THE CHARACTER TO PRINT.

                 IF CRLF
0157 327BF2      STA LASTC+OFFSET ;SAVE CHAR FOR NEXT ENTRY.
                 ENDIF

015A CD6EF2      CALL APPLE+OFFSET ;START 6502 TO PRINT IT.
015D 3A36F0      LDA A$CSW   ;GET CSWL AND SAVE FOR THE NEXT CALL.
0160 3279F2      STA LASTL+OFFSET
0163 E1          POP H  ;RESTORE OLD CSWH AND CSWL.
0164 2236F0      SHLD A$CSW

                 IF ECHO
0167 78          MOV A,B  ;GET SAVED 2ND CHARACTER.
0168 3201F4      STA A$TUBE   ;AND PUT IT BACK ON SCREEN.
016B 7A          MOV A,D  ;RESTORE CH.
016C 3224F0      STA A$CH
016F 7B          MOV A,E  ;RESTORE CV STUFF.
0170 215BFB      LXI H,A$TABV ;WITH MONITOR'S HELP.
0173 CD6EF2      CALL APPLE+OFFSET
                 ENDIF

0176 C9          RET    ;DONE WITH ONE CHARACTER.

                 ; SUBROUTINE TO START 6502 AT (H,L) AND PASS IT THE A REGISTER.....
```

A SOFTCARD LST DRIVER

```
0177 3245F0     APPLE: STA A$ACC  ;SET 6502'S A REG.
017A 22D0F3      SHLD A$VEC   ;SET 6502'S STARTING ADDRESS.
017D 2ADEF3      LHLD Z$CPU   ;FIND SLOT OF SOFTCARD.
0180 77          MOV M,A   ;AND TURN Z-80 OFF.
                    ;CONTINUE WHEN 6502 DOES A RTS.
0181 C9          RET    ;RETURN TO CALLER ABOVE.....

0182 00C1       LASTL: DW 0C100H   ;NEXT PLACE TO ENTER ROM DRIVER.

                 IF CRLF
0184 00         LASTC: DB 0   ;SAVE FOR OUTPUT CHARACTER.
                 ENDIF

0185            LSTEND: END                                    OM DRIVER.


                 IF CRLF
0184 00         LASTC: DB 0   ;SAVE FOR OUTPUT CHARACTER.
```


Highest Quality, **Lifetime Guarantee!**

DISKETTES

$1.65 5¼" soft sectored

- Our own custom-manufactured brand
- Certified for Single & Double Density
- Can be used double-sided on Apple, Atari, IBM, and many other computers
- Hub ring, tight seams, loose-fitting jackets, white Tyvek sleeves
  - * order in any quantity from 1 up
  - * $155 for 100 - $149 for bulk pak of 100 (sleeves & labels separate, write-protect notch on one side)
  - * add $2 to total order for shipping

Hard plastic stand-up 10-diskette carrying cases $2.75 each, 4 for $10 (specify choices: beige, black, blue, green, grey, red, yellow)

Smoked-plastic jumbo-size flip-top 75-diskette file cases $19.50 each

---

VF ASSOCIATES

6327 Western Ave., N.W., Wash., D.C. 20015

(202) 363-1313

Circle 42 on Reader Service Card

Multi-Lingual word processor

For the Apple II+ and Apple IIe

Accented and upper/lower case letters appear on the screen and are easily printed with just about any printer. No extra hardware needed. ONLY $69.95

Spanish	á	é	í	ó	ú	ü	ñ	Ñ	¿	¡
French	é	è	ê	ç	ù	î	ô	â	û	à
English	{	}	[	]	|	\	`			
German	ä	ö	ü	Ä	Ö	Ü	ß			
Italian	à	è	ì	ò	ù	À	È	Ì	Ò	Ù
Danish/Norwegian	å	æ	Å	Æ	ø	Ø				
Finnish/Swedish	ä	ö	Ä	Ö						
Hawaiian	á	é	í	ó	ú	Á	É	Í	Ó	Ú
Hungarian	á	é	í	ó	ö	ő	ú	ü	ű	

See your dealer or send check, M.O. or C.O.D. Add $2.00 for shipping.

THE PROFESSOR Box 301B, Swanton, VT 05488 [514] 747-9130

FREE BROCHURE

Circle 34 on Reader Service Card

FROM THE HEART OF THE MONITOR:

AND NOW, THE WINNER . . .

Ken Kashmarek

IN a recent issue, a question was posed as to why the &MOVE routine is faster than the Monitor move routine for moving bytes in memory. Or is it? If you have a different answer, write and let me know.

First of all, we are only concerned with the 31 bytes of code that actually moves the data and eventually returns control to the user program. The MONITOR MOVE listing provides the Monitor routine, while the FASTER MOVE listing provides the &MOVE routine. Without going into instruction timings, we should be able to determine which routine is faster.

Strictly speaking, the Apple][is limited by the speed of the 8-bit bus. Every instruction accesses at least one byte from memory. Depending on the instruction, more instruction bytes may be fetched, followed by fetch and/or store of data bytes. The bus speed is a major factor in determining how fast data can be moved to and from memory. It also determines how fast instruction bytes can be obtained from memory to be executed. The comparison will be simple: count the total number of memory accesses, including instructions, in order to move a block of data in memory.

The comparison will be for a move of 1000 bytes of memory. Therefore, each routine will cycle 1000 times in order to load (LDA) and store (STA) 1000 bytes. First, the Monitor move routine.

Beginning in line 6, we must fetch two instruction bytes and one data byte (total of three bytes). Line 7 fetches two instruction bytes and stores one data byte (three bytes, running total 6 bytes). Line 8 fetches three instruction bytes, and places two bytes of program counter on the stack (five bytes, running total 11 bytes). Line 9 has two instruction bytes (running total 13 bytes). This sequence of instructions is executed 1000 times for a total of 13000 bytes moved to and from memory (so far).

Line 12 has two instruction bytes, loads one data byte, increments it, and stores the data byte back in memory (total four bytes). Line 13 has two instruction bytes while line 14 uses four bytes (running total 10 bytes). Lines 15–18 use three bytes each (running total 22 bytes). Line 19 uses four bytes, line 20 uses two bytes, and line 21 uses four bytes (running total 32 bytes). Line 22 is one instruction byte, but must fetch two bytes from the stack for the program counter (three bytes, new running total of 35 bytes). In this sequence, INC A4+1 and INC A1+1 are only executed four times to move 1000 bytes, so the total byte count here is (35 - 8) * 1000 + 4 * 8 = 27032.

The total number of bytes (instructions and data) for the Monitor move is 40032 to move 1000 bytes of data.

The &MOVE routine is analyzed in a similar fashion. For the instructions that are executed every time through the loop, there are 29 instruction and data bytes, for a total of 29000. For the instructions that execute only when pointers cross page boundaries, there is a total of 16 * 4 = 64 instruction and data bytes (lines 10, 11, 16, 19, and 29). The overall number of bytes (instruction and data) is 29064 for the &M0VE routine.

On a comparison of the number of bytes moved to and from memory, the &MOVE routine makes 27% fewer memory accesses, and is faster than the Monitor move routine. The big difference is the JSR and RTS for each byte moved by the Monitor routine. This adds 8000 bytes of instructions alone to the loop (four instruction bytes, plus storing two bytes of program counter on the stack for the JSR, and fetching two bytes of program counter from the stack for RTS).

```
:ASM
                          1     * MONITOR MOVE ROUTINE
                          2     A1      EQU   $3C
                          3     A2      EQU   $3E
                          4     A4      EQU   $42
                          5             ORG   $FE2C
FE2C: B1 3C               6     MOVE    LDA   (A1),Y    ;GET A BYTE
FE2E: 91 42               7             STA   (A4),Y    ;PUT A BYTE
FE30: 20 B4 FC            8             JSR   NXTA4     ;INCR & TEST
FE33: 90 F7               9             BCC   MOVE      ;FOR MORE
FE35: 60                  10            RTS
                          11            ORG   $FCB4
FCB4: E6 42               12    NXTA4   INC   A4        ;A4,L
FCB6: D0 02               13            BNE   *+4
FCB8: E6 43               14            INC   A4+1
FCBA: A5 3C               15            LDA   A1        ;NOW SET CARRY
FCBC: C5 3E               16            CMP   A2        ;COMPARE
FCBE: A5 3D               17            LDA   A1+1      ;HI BYTE
FCC0: E5 3F               18            SBC   A2+1
FCC2: E6 3C               19            INC   A1        ;A1,L
FCC4: D0 02               20            BNE   *+4
FCC6: E6 3D               21            INC   A1+1
FCC8: 60                  22            RTS

--End assembly--

31 bytes

Errors: 0
```

AND NOW, THE WINNER . . .

This comparison ignores data movement within the 6502 processor itself. For example, JSR increments the S register (stack pointer) while RTS decrements the S register. Operations internal to the 6502 do not require a memory access, and were not counted. However, those operations do take time and may delay access to memory while the 6502 processor waits for a memory cycle to complete. I expect that a detailed timing comparison would indicate essentially the same result. This comparison was done without the aid of any timing charts. Now, let's hear from the experts out there. Imagine, 30 to 40 thousand bytes moving through the 6502 in order to move 1000 bytes of data in memory.

```
:ASM
                        1     * FASTER MOVE ROUTINE
                        2     SOURCE   EQU  $3C
                        3     TARGET   EQU  $42
                        4     LINNUM   EQU  $50
                        5              ORG  $34F
034F: C6 50             6     MVE0     DEC  LINNUM      ;DEC count
0351: A5 50             7              LDA  LINNUM
0353: C9 FF             8              CMP  #$FF        ;WRAP?
0355: D0 04             9              BNE  MVE1
0357: C6 51            10              DEC  LINNUM+1
0359: 30 12            11              BMI  MVE3        ;Done...
035B: B1 3C            12     MVE1     LDA  (SOURCE),Y  ;Source
035D: 91 42            13              STA  (TARGET),Y  ;Target
035F: E6 3C            14              INC  SOURCE      ;to next byte
0361: D0 02            15              BNE  MVE2
0363: E6 3D            16              INC  SOURCE+1
0365: E6 42            17     MVE2     INC  TARGET      ;TO NEXT BYTE
0367: D0 E6            18              BNE  MVE0
0369: E6 43            19              INC  TARGET+1
036B: D0 E2            20              BNE  MVE0        ;Die at 64K
036D: 60               21     MVE3     RTS

--End assembly--

31 bytes

Errors: 0
```


DISK DRIVES

$199

First time under $200!

At last you can afford to buy another drive!!

- Siemens drive mechanism (uses lead screw instead of rotating cam, for very quiet, reliable operation)
- Works with your present controller
- Can be used as 35 or 40 track drive
- Improved self-centering of diskettes
- Externally-adjustable speed control
- Handles synchronized tracks
- Manufacturer's 1-year warrantee

These superb ready-to-use drives come with standard size and color casings. *Disk controller cards available: $65.00*

No charge cards or COD orders. Add $5 for shipping and handling. Send money order or certified check for immediate shipment, $4 extra for 2nd day delivery
Ask for free 74LS125 chip with order.

VF ASSOCIATES
6327 Western Ave, N.W., Wash., D.C. 20015
(202) 363-1313

Circle 43 on Reader Service Card

FOR APPLE II PLUS, FRANKLIN, APPLE IIe

Font DownLoader*

Expand the capacity of your printer hundreds of times

Load custom fonts into your Apple® Matrix Printer, Prowriter™ 8510A, OKI® Microline 92, 93, 84 Step II, and Epson® FX and use them with virtually every word processor to turn your printer into a **custom typesetter.** After the fonts are loaded, they will stay in your printer until it's turned off. A font editor is also provided to allow you to create your own graphics, text, foreign language letters, math and electronics symbols to load into your printer.

On-Disk *(Specify Printer)* $39.00

*Requires printer interface that adheres to Apple protocall (Tymac, Apple, Epson, etc.) (Will not work with Pkaso, Wizard and some others).

$100 REWARD

Submit the best or most unique font using the above software and we will make you $100 richer. Other prizes for the first 25 runners up.

Dealer and Distributor Inquiries Invited

MasterCard VISA

MICRO WARE D I

1342 B Rt. 23, Butler, NJ 07405
201-838-9027

Circle 24 on Reader Service Card

SYSTEM.LIBRARY

David W. Walker

HERE is another short tip for Apple Pascal programmers who are short of room on their system (boot) disks. That is a condition which tends to plague anyone who has accumulated a large SYSTEM.LIBRARY file, and it is aggravated if you use, as I now do, the Advanced System Editor offered by Volition Systems [see Note]. That editor is magnificent — a must-have for anyone who writes Pascal textfiles regularly — but it takes up 100 blocks. It doesn't have to be on the boot disk; but that is the most convenient place for it if you have fewer than three disk drives

Anyway, back to the main point. One way to make room on your boot disk is to take the file SYSTEM.APPLE off it. That file is needed only during the initial boot, and just sits there hogging valuable disk space after that. On booting, as those of us who started with only one disk drive will remember, SYSTEM.APPLE loads the P-code interpreter, then looks for the file SYSTEM.PASCAL. If it can't find it, it displays a message asking for the file and waits for a Reset!! More on that point later.

So, I have moved SYSTEM.APPLE to the *back* of my boot disk. I know, in theory that's a no-no, using both sides of the disk, but it works for me. I boot from that side, then flip the disk over to the working side, press Reset, and complete booting the Pascal system.

At least that's that's what I used to do until last week, when I replaced my trusty Apple][with my new Apple *//e*. I soon discovered that pressing Reset after SYSTEM.APPLE was loaded does not have the desired effect on an Apple *//e*. Instead of proceeding to load SYSTEM.PASCAL, it reboots the machine! What to do? My boot disk (working side) simply has no room to put SYSTEM.APPLE back on it. Without SYSTEM.APPLE and SYSTEM.PASCAL on the same side of the same disk, it looked as if I wouldn't be able to boot Pascal at all!

You have probably guessed the next step: go ahead and put SYSTEM.PASCAL on the back of the boot disk, with SYSTEM.APPLE. Okay, that allows a boot; but SYSTEM.PASCAL has to be available continuously while I'm using Pascal, especially when I have to keep swapping in and out parts of SYSTEM.PASCAL and it expects to find each part in the same place on the disk where it was when the system was booted.

Solution: Put SYSTEM.PASCAL on *both* sides of the boot disk, in the same place. For example, SYSTEM.APPLE could be the first file in the directory on the "boot" side (or the separate "first boot" disk, if you're skeptical about using both sides). It will occupy 32 blocks, from block 6 through block 37. Then SYSTEM.PASCAL could start at block 38 on that side. Back on the "working" side, the first file could be SYSTEM.FILER, which would fill from block 6 through block 34. The second could be SYSTEM.MISCINFO — one block if you're using the standard editor; two blocks if you're using ASE. Then, using the M(ake function in the Filer, you can create a dummy file (I call mine SYSTEM.FILLER) to occupy blocks 36 and 37 (or block 37 only, if you use ASE). Then SYSTEM.PASCAL can start at block 38, just as it does on the "boot" side.

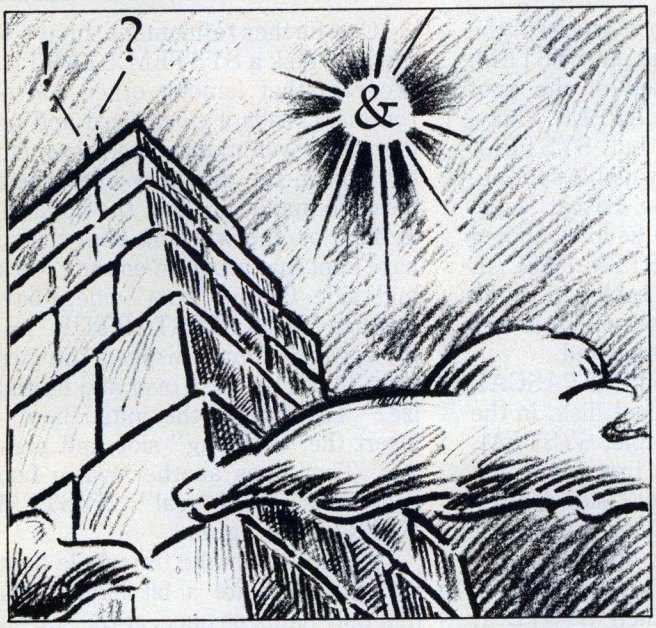
As a further precaution, I have named both the "boot" side and the "working" side of my disk with the same volume name. Now, when I turn the disk over to the working side after the initial boot, the system doesn't know I've done it. It still finds SYSTEM.PASCAL exactly where it expects to find it, and everything works as it should.

One further refinement. I have on my boot disk a SYSTEM.STARTUP program that (among other startup routines) reads a clock card and updates the system date, both in memory and in the directory of the boot disk. To make sure that the updating is done on the working boot disk, I have kept that program on the "working" side. On the "boot" side, I have another SYSTEM.STARTUP program which simply displays the message "Turn this disk over and I(nitialize." Then, after the initial boot, I insert the "working" side and press "I." That does another warm boot and runs the "real" SYSTEM.STARTUP program.

Yes, it does make the whole booting process quite a bit longer. But that only happens once, at the start of a session; and the extra room made available for a *big* SYSTEM.LIBRARY file, plus a SYSTEM.STARTUP program, plus the Advanced System Editor, makes the rest of each programming session far more convenient and comfortable.

NOTE: Advanced System Editor is available for $175, postage paid, from Volition Systems, P.O. Box 1236, Del Mar, California 92014. That's a lot of money; but it's worth it if you spend a lot of your time programming in Pascal. Among other nifty features, ASE can edit files longer than 40 blocks (in fact, it can edit any length of file, up to the capacity of the disk); it automatically backs up a file before editing it; and it allows user-defined macros and user-defined key definitions (so that, for example, it can be configured to recognize the up-arrow and down-arrow keys on the Apple *//e* in addition to the usual [Ctrl-O] and [Ctrl-L] keys).

Trapped by the language? They found out in Babel what happens when your language won't do what you want it to.



With AMPER-MAGIC™ there's no need to know machine language to benefit from its power.

To your Applesoft™ programs you can attach, finished machine language routines (referenced by name, not by address) in seconds. No separate BLOADing. And with AMPER-MAGIC commands, you can pass variables back and forth directly, just as with built-in Applesoft commands.

So, name your routine, perform the append procedure, and AMPER-MAGIC takes over — machine language speed where it counts: in *your* programs. Even your commercial programs.

AMPER-MAGIC COMMAND LIBRARIES™ gives you access to hundreds of relocatable machine language routines, including those from magazines and other libraries, and - most notably - those from AMPER-MAGIC itself, which has over fifty routines of its own in two volumes, with more to come.

Imagine!
Only $75.00 for the original AMPER-MAGIC by Bob Nacon
Only $35.00 for AMPER-MAGIC COMMAND LIBRARY™, Volume 2

Plus shipping. MC and Visa accepted.

Amper-Magic™

See your dealer or contact us.
Anthro-Digital. 103 Bartlett Avenue, Pittsfield, MA 01201
Telephone (413) 448-8278, Telex 467622
"The Company That Interfaces People and Computers"

For those of you who would like AMPER-Magic's routines in your own programs, just ask for Anthro-Digital's no cost licensing policy.

AMPER-MAGIC and AMPER-MAGIC COMMAND LIBRARY™, are trademarks of Anthro-Digital, Inc.
Applesoft is a trademark of Apple Computers, Inc.

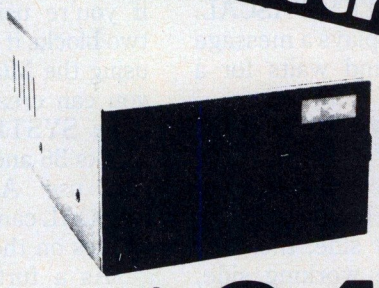
Also available from Anthro-Digital: Versacalc (several versions), Visicalc Formatting Aids, The Executive Secretary, The Personal Secretary, and Executive Speller, Hebrew II and Hebrew II Plus, S-C Macro Assembler, QuickTrace, Amper-Magic, The Rental Manager, F.A.R.M. accounting packages, The Performance Manager, Omniscan VideoDisc interface, and Flipper. **DEALERS INVITED**

Circle 8 on Reader Service Card


```
10  REM

SO WHAT DID YOU EXPECT
    BY RUSS STILL

  CALL -A.P.P.L.E.

15  POKE 232,2: POKE 233,3:
20  DATA 1,0,4,0,2,4,2,4,0,6
30  FOR I = 1 TO 10: READ F: POKE
     769 + I,F: NEXT
50  HGR2 : SCALE= 90: HCOLOR= 3
60  FOR R = 1 TO 255: ROT= R: XDRAW
     1 AT 140,90: NEXT
70  GOTO 60
```

```
10  REM
SO WHAT DID YOU EXPECT
    BY RUSS STILL

  CALL -A.P.P.L.E.

15  POKE 232,2: POKE 233,3:
20  DATA  1,0,4,0,2,4,2,4,2,4,2,4,0,6
30  FOR I = 1 TO 14: READ F: POKE
     769 + I,F: NEXT
50  HGR2 : SCALE= 40: HCOLOR= 3
60  FOR R = 1 TO 64: ROT= R: XDRAW
     1 AT 140,90: NEXT
68 J = J + 1: POKE 776,J
69  IF J > 254 THEN J = 0
70  GOTO 60
100  REM "

The two programs demonstrate
an interesting anomaly.
A single "XDRAW" causes a line
to be displayed which
immediately disappears
```


Laing Electronics



μ-SCI
MICRO-SCI
A2 DRIVE

$249

BUY THE FIRST AND THE BEST APPLE COMPATIBLE DRIVE!

A2 DRIVE WITH CONTROLLER **$309**
(FREE SHIPPING IN USA OR CANADA.
A2 MODELS ONLY.)

A40 (40 TRACK DRIVE W/O CONTROLLER)	**$284**
A70 (70 TRACK DRIVE W/O CONTROLLER)	**$365**
— CONTROLLER FOR A40 OR A70 DRIVES	**$ 79**
A3 APPLE III DIRECT REPLACEMENT	**$299**

10382 Stanford Ave.,E-203, Garden Grove, Cal., 92640
Phone (714) 534-4216

Check or Money Order, — NO CREDIT CARDS! Check's Allow 10 BANKING DAYS FOR CLEARANCE. Prices and specifications subject to change without notice. Add $7.50 for UPS surface. Cal. Res. add 6% Sales Tax.

Circle 19 on Reader Service Card

PASCAL 1.2

Allen Munro

USING Apple Pascal with the Apple *//e* presents a number of problems, especially for owners of single-drive systems. Many of these will be obviated by Apple's new Pascal operating system for the Apple][to be made available in the fourth quarter of this year. For one thing, the old two-stage boot using APPLE3: and APPLE0: will be possible on a *//e*. How? The solution is very simple. The *//e* reset key causes a cold boot to be performed. The new Apple Pascal 1.2 will simply use a different key such as [Return] to initiate the second stage of the boot, instead of [Reset].

It will also be unnecessary to use the SETUP program to change SYSTEM.MISCINFO to reflect the new up-arrow and down-arrow keys for the *//e*. Instead, the operating system will automatically translate these keys into the old [Ctrl-O] and [Ctrl-L]. It will therefore be possible to use both the old Apple][Pascal control keys for cursor movements and the new Apple *//e* arrow keys interchangeably.

These two changes are only a few of the features that are designed to provide all Apple][Pascal users — original Apple][, Apple][Plus, and Apple *//e* owners — with a Pascal operating system that can be used in the same way. Superficially, the new operating system will not differ much from the old 1.1 operating system (or its predecessor, 1.0, which was originally called 11.1), but it will fix a number of undesirable features. One example is that the new disk driver will not disable interrupts. (This problem is patched up in certain commercial packages, such as Bill Blue's P-Term, but it will be nice to have an official fix from Apple.)

Apple has identified 39 bugs in Apple Pascal 1.1 that will be repaired in 1.2. Furthermore, the new Pascal operating system will provide full support for the Apple *//e*. Systems equipped with the extended memory 80-column card will be fully supported in the new operating system. Data and native code (assembly language portions of programs) will reside in the original Apple][memory area, while P-code will be in the extended memory card. While this increase in size will probably carry some execution speed penalties, it should permit the construction of much larger Apple][Pascal programs.

128K-byte Apple *//e*'s will be able to have segment members up to 63, and segment libraries. The Unitstatus (2) function will tell whether the new open-apple and/or closed-apple and/or shift keys are pressed. The list of new features goes on, and expert Apple Pascal programmers are probably looking forward eagerly to solutions to particular programming problems they've encountered.

Novice Apple Pascal programmers should not be intimidated by the esoteric new features. There is a good chance that the differences will be in no way apparent in most Pascal programs. Apple will undoubtedly make the upgrade available inexpensively to old Pascal system owners. So take heart, *//e* owners using Pascal! The best is yet to come.

A.P.P.L.E. members only!

The A.P.P.L.E. Applesoft Carpenter //

by Wayne Eastwood

AVAILABLE NOW

with new features and extended parameters

- LISTER: a program list formatter
- COMPARER: compares two programs for differences
- CRUNCHER: compacts programs, strips REMs
- CONVERTER: converts Integer to Applesoft, corrects all syntax — Integer Basic not required in system.
- CONSTRUCTER: saves text file image of Integer program for future reference.
- FINDER: locates any string, token or variable and lists lines
- KEEPER: saves variables while editing program lines

Fast machine language routines

INTRODUCTORY PRICE

$18.50

Overseas shipping weight: 2.5 oz.
product code SCAR

All foreign orders (including Canada) will require additional postage and shipping.

24 hour toll-free order no.

800-426-3667

(except in Washington)
Washington state residents add 7.9% sales tax.

A.P.P.L.E. Orders
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245
Visa, M/C welcome

APMAIL APPLE-CATIONS

Ken Kashmarek/Call —A.P.P.L.E. staff writer

Fast–Loading APMAIL

LATE one night, while devouring In Depth #3: All About DOS, I came across an Editor's Note on page 229. The note said that COMPLEAT ZAP will take up to 60 seconds to load the first time it is run, and that FAST LOAD FORMATTER (from page 61) would reduce this time to about 11 seconds. Val was right! What a difference. Now, to find something else to use FAST LOAD FORMATTER on.

I had just obtained a copy of Apmail Version 3.0 to handle my mailing lists. The largest program on the disk, GENERAL APMAIL FUNCTIONS, is 89 sectors long, and takes over 20 seconds to load. The perfect candidate! Boy, was I wrong.

Apmail has six program modules, ranging from 16 sectors to 89 sectors in length. The HELLO program usually invokes GENERAL APMAIL FUNCTIONS which in turn invokes one of the four auxilliary programs to perform various functions. The four auxilliary programs always return to HELLO. Much of the time in Apmail is spent switching between programs.

So, out came FAST LOAD FORMATTER, and a copy of the Apmail diskette. All six programs were converted, and HELLO came up just fine. The faster load time for HELLO was immediately obvious. The auxilliary programs all worked from HELLO, but GENERAL APMAIL FUNCTIONS did not.

It turns out that HELLO invokes GENERAL APMAIL FUNCTIONS via the CHAIN program, in order to pass along an initialized variable table and set of arrays (CHAIN is described on page 106 of the DOS manual). However, the fast loader program set up by FAST LOAD FORMATTER does not pass along the variable table as set up by HELLO. Some changes are necessary in order to pass the variable table along to GENERAL APMAIL FUNCTIONS.

FAST LOAD builds two modules. The fast loader module is only 2 sectors long. When HELLO calls CHAIN, the fast loader program is loaded. The variable table is passed to the fast loader program. After the fast loader has loaded the APMAIL program, it then calls the Applesoft CLEAR function. BIFF! SOCK! POW! The variable table just hit the bit bucket.

Without further changes, FAST LOAD can be used on all of the Apmail programs, except GENERAL APMAIL FUNCTIONS. This works fine, except you must put up with the long 89 sector DOS load time. We can make some changes to use fast load with the main Apmail routine.

In order to preserve the variable table passed from HELLO to APMAIL, it must be built at an address high enough in memory to load the 89 sector APMAIL program below it. This is done by telling FAST LOAD FORMATTER to load HELLO at $5F01 (the variable table must be above $5E11, and FAST LOAD requires the low byte of the load address to be $01 — therefore, $5F01, the next page above $5E11). In order for HELLO to run, line 31 must be changed. "IF A < 22000 + TT . . ." must be changed to "IF A < TT . . . " (remove "22000 + ").

FAST LOAD FORMATTER should set up GENERAL APMAIL FUNCTIONS to load at $0801 (the default). Before booting the APMAIL disk, four changes must be made to the fast loader program for GENERAL APMAIL FUNCTIONS. Use the following sequence of commands:

```
LOAD GENERAL APMAIL
  FUNCTIONS
CALL -151
840:EA EA
844:EA EA
84E:20 97 D6
855:20 80 D6
3D0G
SAVE GENERAL APMAIL
  FUNCTIONS
```

The first two changes prevent the variable table address from being changed ($69/6A). Thus, the variable table pointer passed along from HELLO by the CHAIN program is left intact. The third change eliminates the call to the Applesoft CLEAR function. Instead, the routine which updates TXTPTR is called. The fourth change then replaces 3 NOP bytes with a call to the Applesoft stack initialization routine (part of CLEAR). Remember that the program which is being changed is actually the fast loader program. The actual Applesoft program that is loaded by the fast loader has the same file name followed by an asterisk.

With these changes complete, you can now boot your modified Apmail diskette and realize an eight second load of GENERAL APMAIL FUNCTIONS, and faster loads for HELLO and the four auxilliary programs. This significantly improves the usability of Apmail. I have applied this technique to Apmail Version 3.0 as it is distributed. I have also applied it to a modified version that I use for my own files. If you make program modifications, the load address for HELLO may have to be changed. I did not attempt to update FAST LOAD FORMATTER to automatically make these changes.

Some cautions are in order. FAST LOAD will not work with DOS in the RAM card. It may also fail with many of the brand X improved or modified versions of DOS that are available. Make these changes to a copy of your Apmail diskette (never to the original or your only copy). After making a copy of your Apmail diskette, delete the six Apmail programs. When running FAST LOAD FORMATTER, use your original diskette as input and your copy as output for the updated programs. Otherwise, you will run out of sectors on the output diskette (FAST LOAD needs consecutive sectors for each of the programs).

Multiple APMAIL Files

Apmail version 3.0 supports one mailing label file set: TITLE, NAMES, and NAMES INDEX. By using EXEC files, several different mailing label file sets can be used. The EXEC file contains the sequences of commands for UNLOCK, RENAME, and LOCK to switch from one set of files to another.

To convert from APMAIL file names to a different set of file names, I use EXEC file AP TO GENERAL which contains these commands:

```
UNLOCK TITLE
UNLOCK NAMES
UNLOCK NAMES INDEX
RENAME TITLE,GENERAL.TITLE
RENAME NAMES,
  GENERAL.NAMES
RENAME NAMES INDEX,
  GENERAL.NAMES INDEX
LOCK    GENERAL.TITLE
LOCK    GENERAL.NAMES
LOCK    GENERAL.NAMES INDEX
```

This mailing label file set exists on the same disk with APMAIL. GENERAL is the mailing list name. To convert these files back to the names which must be used by APMAIL, I use EXEC file GENERAL TO AP which contains these commands:

```
UNLOCK GENERAL.TITLE
UNLOCK GENERAL.NAMES
UNLOCK GENERAL.NAMES INDEX
RENAME GENERAL.TITLE,TITLE
RENAME GENERAL.NAMES,
  NAMES
RENAME GENERAL.NAMES INDEX,
  NAMES INDEX
LOCK    TITLE
LOCK    NAMES
LOCK    NAMES INDEX
```

To convert my Christmas list for use with APMAIL, I use EXEC file CHRISTMAS TO AP, which contains these commands:

```
UNLOCK CHRISTMAS.TITLE
RENAME CHRISTMAS.TITLE,TITLE
LOCK    TITLE
```

In this case, only the TITLE file is on the same disk with APMAIL. The other files are on a separate disk and do not have to be renamed. To convert the TITLE file back so I can again use the GENERAL file, EXEC AP TO CHRISTMAS which contains these commands:

```
UNLOCK TITLE
RENAME TITLE,CHRISTMAS.TITLE
LOCK    CHRISTMAS.TITLE
```

Any number of address lists can be maintained this way. Of course, the command sequences could also be programmed into Applesoft with the mailing label names (GENERAL or CHRISTMAS) input to the program.

```
]LIST

10  REM

APMAIL ][ VERSION 3.0
   FILE CONTROLLER
  BY  KEN KASHMAREK

  CALL  -A.P.P.L.E.

30  TEXT : HOME : HTAB 9: INVERSE
    : PRINT "APMAIL FILE CONTROL
    LER": NORMAL
40 LO$ = "LOCK    ":UN$ = "UNLOCK
    ":RE$ = "RENAME "
50 TI$ = "TITLE":NA$ = "NAMES":NI
    $ = "NAMES INDEX"
60 O$ = "OPEN ":R$ = "READ ":W$ =
    "WRITE ":C$ = "CLOSE ":V$ =
    "VERIFY ":D$ =  CHR$ (4):MO$
    = "MON C":NO$ = "NOMON C"
70  ONERR  GOTO 100
80  PRINT D$;V$;TI$: PRINT D$;O$;
    TI$: PRINT D$;R$;TI$: INPUT
    F$,S$,T$,S,D,V,NS,ND,NV: PRINT
    D$;C$;TI$
90  GOTO 110
100  POKE 216,0:F$ = "(NONE)"
110  PRINT : PRINT "CURRENT FILE
    IS: ";F$
120  PRINT : INPUT "SWITCH, RENAM
    E, OR QUIT (S/R/Q)? ";Q$
130  IF Q$ <  > "S" OR F$ = "(NON
    E)" THEN 260
140  PRINT D$;MO$
150  PRINT D$;UN$;TI$
160  PRINT D$;RE$;TI$","F$"."TI$
170  PRINT D$;LO$;F$"."TI$
180  IF S <  > NS OR D <  > ND THEN
    250
190  PRINT D$;UN$;NA$
200  PRINT D$;UN$;NI$
210  PRINT D$;RE$;NA$","F$"."NA$
220  PRINT D$;RE$;NI$","F$"."NI$
230  PRINT D$;LO$;F$"."NA$
240  PRINT D$;LO$;F$"."NI$
250  PRINT D$;NO$
260  IF Q$ = "Q" THEN 440
270  PRINT : INPUT "NEW FILE IS:
    ";N$: IF N$ = "" THEN 440
280  ONERR  GOTO 310
290  PRINT D$;V$;N$"."TI$: PRINT
    D$;O$;N$"."TI$: PRINT D$;R$;
    N$"."TI$: INPUT X$,S$,T$,S,D
    ,V,NS,ND,NV: PRINT D$;C$;N$"
    ."TI$
300  GOTO 320
310  POKE 216,0: PRINT "NEW FILE
    NOT FOUND": END
320  PRINT D$;MO$
330  PRINT D$;UN$;N$"."TI$
340  PRINT D$;RE$;N$"."TI$","TI$
350  PRINT D$;LO$;TI$
360  IF S <  > NS OR D <  > ND THEN
    430
370  PRINT D$;UN$;N$"."NA$
380  PRINT D$;UN$;N$"."NI$
390  PRINT D$;RE$;N$"."NA$","NA$
400  PRINT D$;RE$;N$"."NI$","NI$
410  PRINT D$;LO$;NA$
420  PRINT D$;LO$;NI$
430  PRINT D$;NO$
440  END
```

A Call —A.P.P.L.E. SOFTWARE REVIEW:

THE GAMING ROOM

Val J. Golding

WE don't know what the attraction is. Why is it someone (and not just kids) will sit at a computer for hours on end, playing Wizardry or Space Invaders. Certainly there is a compulsive element involved. Alcoholism is now scientifically established as a medical condition, disease if you will. An alcoholic often knows when he/she should quit, but cannot.

Las Vegas and the state of Nevada have their share of compulsive gamblers, those individuals who, in one form or another, have wagered and lost more than their budget permits, yet refuse to quit when "the cards are down." These people will wire home for more money, hock their watch, anything for one more chance at the "big win." Evidence indicates the possibility that this too may actually be a medical problem. This is a logical assumption, although we can not at the moment support it.

There exists an even closer relationship between gambling and gaming than there does between drinking and gambling. Many newspaper features and magazine articles of late have addressed arcade game addiction, particularly among teen-agers. The desire to win in gaming is much akin to the challenge of hitting it big in gambling. We can therefore reasonably conclude that the causes are closely related.

This is not to say that arcade games per se are bad. Quite the contrary. We believe they have much to contribute in terms of improved coordination and logic stimulation. Problems occur only when, like alcohol, it is allowed to overpower and control your day to day activities. We find arcade and adventure games a panacea. We can sit at our Apple and think of the projects facing us that we would just as soon not do, and end up by playing a couple hours worth of games. Having done so, particularly if we did not allow the dumb game to get the best of us, we can quit, basking in the glow of success, and cheerfully proceed to our assigned tasks.

Life is such a bowl of cherries, particularly if they are three or four abreast!

Jawbreaker *$29.95*
Sierra On-Line, Inc.
36575 Mudge Ranch Road
Coarsegold, CA 93614
Copy protected

The original Pacman by Atari was a real winner, the "All in the Family" of video games, producing probably more spin-offs and look-alikes than the noted TV series, along with numerous lawsuits. Snack Attack by Datamost was among the earliest of these and probably was too close to call. The law in this area is much too vague, although Atari seems to have made out pretty well.

The basic premise of all Pacman genre games, in the surroundings of multiple mazes, is eat or be eaten. Whether the former or the latter occurs is largely the result of how recently the protagonist has consumed an energy capsule, which then temporarily allows the attackee to attack the attacker. By consuming all the goodies in the maze, one can progress to the next maze and start all over again. The level of difficulty and often the speed gets tougher with each subsequent maze.

Jawbreaker differs in the main by discarding the maze concept outright. Instead, the screen is divided into five horizontal areas, populated by four jawbreakers and a monstrous pair of teeth. The divisions are physical barriers with a small opening, each moving back and forth at a rate deliberately unsynchronized to that of the jawbreakers, making for constant split-second decisions. The user can select joystick, joyport or keyboard, and in the latter case can choose any four key combination desired.

Since high scores are recorded for posterity on the diskette, this game has been the subject of inter-office rivalry between ourselves and Kathy Pride. We won't disclose who currently holds the number one spot, suffice to say it takes considerable skill and practice to build scores to the 40,000 plus point range. This is one of those "betcha can't play just one more" games and comes with a "try it, you'll like it" rating from this reviewer.

Spy's Demise *$19.95*
Penguin Software
830 4th Avenue
Geneva, IL 60134
Copy Protected
Backup $5 after 60 days, presumably free till then.

Here's another one that can really get your goat. It is fiendishly designed at every level to frustrate your goals, which include sneaking your way up a multi-story building, while avoiding guards in fast moving elevators and at the same time picking up pieces to a secret encoded message, which may or may not save the world. Penguin craftily promises the possibility of a free Spy's Demise tee-shirt for correct solutions, and a hint of perhaps other prizes. That means its gotta be tough, and we agree, having so far attained only slightly more than half of the floors in the first building.

Yes, we said the first building. Should you be so fortunate as to work your way to the top and retrieve the secret message (or at least part of it), you will be plopped into a different, and shorter building. Regretably, the number of elevator guards does not decrease proportionately, thus the skill required to continue is that much greater.

For those of you, who like ourselves, are buffaloed by multiple action keys, Spy's Demise is a joy (pun intended) to play. If you do not use a joystick, the two arrow keys are the only ones required to move the spy in the course of his upward journey. This is an ingenious game that can take considerable time to gain the necessary knowledge of when to dart between the constantly rising and falling elevators without getting clobbered. It is highly recommended for all except those who frustrate easily.

INVERSE STRING PRINTER

Stowe Keller

From BASIC POKE 775,128 will convert output to flashing. POKE 775,119 will restore INVERSE output.

```
:ASM
                    ********************************
                    *                              *
                    *   INVERSE STRING PRINTER     *
                    *      by  Stowe Keller        *
                    *                              *
                    *      Call -A.P.P.L.E.        *
                    *                              *
                    * This routine is relocatable. *
                    * To initialize, BRUN STRING-  *
                    * PRINTER, which will set up   *
                    * the ampersand vector to      *
                    * point to the routine.        *
                    *                              *
                    * Syntax: &[string expression] *
                    *     eg: &A$                  *
                    *                              *
                    ********************************
                    *
                            ORG   $2F5
                    *
                    AMVEC   EQU   $3F5
                    STRPRT  EQU   $DB3D      ;String print
                    CHKSTR  EQU   $DD6C      ;Make sure last result was a string
                    FRMEVL  EQU   $DD7B      ;Formula evaluate
                    SETNORM EQU   $F273      ;Set NORMAL text output
                    SETINV  EQU   $F277      ;Set INVERSE text output
                    *
02F5: A9 00                 LDA   #<INV_PRT
02F7: 8D F6 03              STA   AMVEC+1
02FA: A9 03                 LDA   #>INV_PRT
02FC: 8D F7 03              STA   AMVEC+2
02FF: 60                    RTS
                    *
0300: 20 7B DD      INV_PRT JSR   FRMEVL     ;Evaluate formula
0303: 20 6C DD              JSR   CHKSTR     ;Check that it's a string
0306: 20 77 F2              JSR   SETINV     ;Set INVERSE
0309: 20 3D DB              JSR   STRPRT     ;Print the string
030C: 20 73 F2              JSR   SETNORM    ;Return to NORMAL text output
030F: 60                    RTS

--End assembly--

27 bytes

Errors: 0
```

```
]LIST

10  REM

INVERSE STRING PRINTER DEMO
     BY  STOWE KELLER

     CALL -A.P.P.L.E.

15  HOME
20  PRINT "INVERSE STRING PRINTER
      DEMO BY STOWE KELLER"
30  REM SET UP AMPERSAND HOOKS:
40  POKE 1013,76: POKE 1014,0: POKE
     1015,3: REM $300
50  GOSUB 500
60  PRINT "THIS IS NORMAL TEXT"
70  & "THIS IS INVERSE TEXT USING
      A LITERAL"
75  PRINT
80 A$ = "THIS IS INVERSE WITH A S
     TRING VARIABLE"
90  & A$
95  PRINT
100  & A$ + " IN A STRING EXPRESS
     ION."
105  PRINT
110  PRINT "SEE HOW EASY ";: & "Y
     OU CAN MIX": PRINT " NORMAL
     ";: & "AND INVERSE": PRINT "
      TEXT?"
499  END
500 X$ = "300:20 7B DD 20 6C DD 2
     0 77 F2 20 3D DB 20 73 F2 60
      N D9C6G"
510  FOR I = 1 TO  LEN (X$): POKE
     511 + I, ASC ( MID$ (X$,I)) +
     128: NEXT : POKE 72,0: CALL
      - 144
520  RETURN
```

HOW I BECAME AN AUTHOR FOR CALL —A.P.P.L.E.

Daniel T. Lyons

SINCE you are reading this, you are probably a lot like me. Three years ago I bought an Apple][computer. Boy was I happy. Since I am a very curious type, I read everything I could get my hands on. The DOS manual, the Integer Basic manual, the Applesoft manual, books on programming the 6502, etc. A friend recommended that I join *Call -A.P.P.L.E.* because of the magazine. Reading the extremely clever solutions to some of the bugs that infect the Apple was inspiring. Finding out the tricks that really push my Apple *beyond* its initial apparent limitations gave me a feeling of — WOW — how did they ever think of that!

In the meantime I was busy learning. I disassembled DOS and tried to figure out what was going on. Several books and articles (in *Call -A.P.P.L.E.* — of course) really helped things along. Figuring out what made other peoples program work helped me learn a lot about programming, especially in machine language. Usually this was motivated by my desire to add my own modifications to make the program "better".

Finally it occurred to me that I knew quite a bit about the Apple. Why couldn't I be one of those geniuses that write articles for that fountain of knowledge known as *Call -A.P.P.L.E.*? Not being able to think of any reason why not, I decided to become a writer.

The first thing I sent in was a response to the "So What Did You Expect?" contest in the September 1981 issue. You may remember it. They gave a program which did a bubble sort on the rows and columns of the Low-Res graphics screen after initializing it with random colors. To give you an idea of the time involved to write that article, add up the time it takes to: (1) type in the program, (2) watch it run several times (it wasn't very fast), (3) modify it so it used "nice" initial patterns rather than random ones (more watching — still slow), (4) modify it some more so it would run faster (more watching), and of course (5) write a glowing description of the program in glorious line-by-line detail (they asked for a tutorial) including my own improvements at the end, (6) sprinkle witty sayings in the text to keep the readers happy, (7) edit it several times so that every word was just perfect (after all not just anybody was going to read this — you were!), (8) mail it to *Call -A.P.P.L.E.* with a cover letter that said nice things about *Call -A.P.P.L.E.* and a disk containing the article.

You don't need your computer to tell you that this took quite a lot of time and effort. Six months went by without a word from *Call -A.P.P.L.E.* (to thank me for my fine article and return my disk). Finally, the winner was announced in the March 1982 issue. If you have perfect recall you realize that the winner of the contest was *not* Daniel T. Lyons (happy-go-lucky first time *Call -A.P.P.L.E.* author) but was Scott Somebodyorother. SO WHAT DID I EXPECT? At least I learned something about bubble sort algorithms.

That was not the way I envisioned launching my career as a microcomputer author/expert. Not only wasn't my entry published, but I didn't even get my disk back. This was a setback, of course, but I don't give up easily as my "new" wife wistfully remonstrates during those times she wonders what she saw in me in the first place.

So I tried again. This time I wrote up a description of a very clever program I wrote that allows me to go from Apple Pie to FORMAT (or back again) in less than a second. This was such a neat modification that there was no way that I'd be rejected this time! I suppose that I can't count the time it took me to figure this one out (have you ever disassembled Apple Pie or FORMAT?) because I did that for myself. Once again I wrote the perfect article (I'm modest if nothing else), and sent it in. I didn't send a disk this time because I was still mad about losing the first one. Joy of joys, a few months later I received a letter from Val Golding himself. "Thank you for your very fine article," he agreed. "Unfortunately we don't support obsolete software..." I think I'll frame it.

What happened was that Apple Pie became the new and improved PIE-WRITER. Sigh. Maybe this won't be a complete loss either. If you send me $5, I will be glad to mail you an autographed copy of the article. Each wonderful page comes with my name and address printed neatly on the bottom — just like the "Author's Guidelines" writeup requests.

Like I said before, I don't give up easily. My next project was a DOS modification that allows you to load programs from standard DOS diskettes at speeds up to 500% faster than normal DOS. I went to a lot of trouble to make it interesting — only three bytes in DOS had to be changed, and the program itself was completely relocatable. As I neared the completion of the article, wonderful rumors appeared in *Call -A.P.P.L.E.* They were coming out with a new book — *Call -A.P.P.L.E.* In Depth #3: All About DOS — that had ways to speed up disk access. Simultaneously, an article appeared in *Softalk* about how DOS could be speeded up by bypassing the DOS buffers and loading sectors directly into memory.

By this time my article (which once again had been polished to perfection) was complete. So with a heavy heart I sent it to *Call –A.P.P.L.E.*, knowing as I did so that I was just wasting another eighty-eight cents postage and some spare time.

This time I received the reply in record time (only two weeks). "Dear Daniel. . .thank you for your fine article," Val said, "Unfortunately, we have just published All About DOS. . ." You can easily guess the rest. I've ordered a copy of this nefarious new book, and I'll bet it's got some great programs in it. None of them are mine. At least this time Val suggested that I send my DOS article to the *Apple Orchard.*

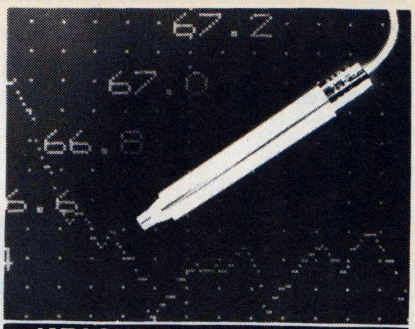
Although I was building my knowledge about the Apple, I was having trouble coming up with an article that *Call –A.P.P.L.E.* would publish. I've had some other ideas — like the joystick made from an applesauce can (it uses 100K ohm pots rather than Apple's standard but hard to find 150K ohm pots). But after I saw the trick with the pots published in *Softalk* (not by me) I gave up on publishing that one. So you see, some of my articles have been squashed before they even made it onto paper.

You probably know the old saying "Frustration is the mother of invention." So in my most inventive mood, I sat down and wrote this article on how to become an author. As you can see from my list of near publications, I am almost an expert on the subject. Each attempt brought me closer to my goal. The first submission didn't receive any response. The next article came back after several months. The last one came back in only two weeks with the positive suggestion that I send it elsewhere for publication. If this trend continues, I've finally hit the jackpot and you are reading this literary gem, if not. . .it's back to the electronic drawing board.

Well, this is all the time I can spare to tell you about what to expect when you send in your first article. I have another hot idea that I have to develop before someone else beats me to it! In the meantime keep the following advice in mind if you *really* want to publish an article in *Call –A.P.P.L.E. — keep trying.* I finally published an article and you can too!

Well, now you did it, Dan, you brought out so many of those things we like to hide! Actually, the points Dan made were pretty accurate. If you followed his story, what it does bring out is that growing up is a slow process, but we're still at it.

There's other good information and hints in his story too. If you want to write for Call –A.P.P.L.E., *its often helpful to contact us beforehand at the editorial office and submit a brief outline of what you have in mind. This is especially true if your proposed article would run more than a couple of typeset pages.*



MEASURE and CONTROL TEMPERATURE

Automatically monitor and record temperature or relative humidity readings with your computer. ADC calibrated temperature probes and measurement software now make it easy. Combine our accurate measurements with time and power interfaces for full closed-loop control.

The Nth Degree system supports 1 to 256 channels with resolution to .01 degree. Standard temp range is -50 to + 105 degrees C. Other ranges are possible. Rugged sensors may be hundreds of feet from your computer without loss of accuracy.

Thermonitor (tm) software provides graphic and numeric display of temperature with data logging on disk or printer. Hooks are provided so that you can customize applications. The Strip-Chart recorder program continuously graphs data on the screen in easy-to-read classic format. Data compression method stores more than 120,000 samples per disk.

American Data Cable, Inc.
2864 Ray Lawyer Drive #205
P.O. Box 2212 • Placerville, CA 95667

1-Channel package with probe and software: **$129.95**
8-Channel package with 1 probe and software: **$189.95**
Strip-Chart recorder and data logger: **$ 49.95**
For VISA and Mastercard call (916) 622-3465

Circle 6 on Reader Service Card

RAM DRIVE EMULATOR
"DRIVE 3"

Use the extra memory in your Apple as a disk drive.

$29.95 **$14.95**

- All DOS commands work the same
- 310 sectors of simulated disk storage with extended 80 col. card
- Incredibly fast
- Full package of utilities
- Fully copyable

VISA Order by phone or mail MasterCard

Software Banc, Inc.
(414) 271-0100

Select version:

☐ Apple IIe $29.95 - 310 sectors, requires extended 80 col. card

☐ 64K Apple $14.95 - 62 sectors, works with any 64K Apple

Name _____

Address _____

City _____

☐ I've enclosed payment. I understand there is no shiping charge for cash or check orders.

☐ Charge my: ☐ Master Charge ☐ Visa

Account # _____ Exp. date ____

☐ Send C.O.D.
Add $1.50 for shipping on charge and C.O.D. orders.

1225 N. Water St., Milwaukee, WI 53202

Circle 3 on Reader Service Card

ACCESS MICRO
11306 Southland Road
Forest Park, Ohio 45240

SOFTWARE FOR YOUR COMPUTER

- **Attractive Prices**
- **Informative Catalog***
- **Prompt Shipping**
- **Toll Free Ordering**

*Our catalog contains useful programming tips -don't miss it !

CALL
1-800-543-1114
Ohio · Alaska · Hawaii
Call 513-825-5803

Circle 4 on Reader Service Card

A Call —A.P.P.L.E. APPLE-CATION:

TELECOMMUNICATING WITH FILEDUMP

Val J. Golding/Steve Ross

A story on automated typesetting appeared in the May *Call -A.P.P.L.E.* which told of efforts in the field by a Minneapolis firm. An accompanying boxed feature briefly described how *Call -A.P.P.L.E.* had been using this technology over the past year and a half.

One major portion of the process involves telecommunicating text files that have been formatted according to the requirements of the typesetting machine. For this we used a D. C. Hayes *Micromodem II* in conjunction with a popular terminal program which would download text files from a diskette and assist in transmitting the data over telephone lines.

Most terminal programs will read the text file from the disk into an Applesoft string array prior to transmission, then output up to 255 characters of each array element to the micromodem. The system worked well, but even at 300 baud, the modem was idle a good deal of the time, waiting to receive more data from Applesoft.

Enter FILEDUMP, written by Lee Reynolds and published in the June 1982 *Call -A.P.P.L.E.*, one of the most popular utilities we have published. Steve Ross, who heads the software development section at our general office, handles the telecommunication of software documentation files to the typesetter. Steve saw in FILEDUMP a means to speed up a slow process and wrote a bare-bones program which utilized FILEDUMP to get data off the disk and pass it on to the modem. This resulted in perhaps a 25% improvement in our transmission time, and when you are paying a typical typesetting rate of $50 per hour, the savings are substantial indeed.

We took Steve's program and added a large number of bells and whistles which made it a little more goof-proof and easier to use by ourselves and our staff, then the thought crossed our mind that here was a practical application that although Call -A.P.P.L.E.-specific, could be used by others with only minor modifications.

```
]LIST

10  REM

          TELEDUMP
BY STEVE ROSS/VAL J GOLDING

CALL -A.P.P.L.E.  AUG. 1983

100  TEXT : HOME : INVERSE : PRINT
     "PRESS RETURN FOR CATALOG, C
     TRL-C TO QUIT": POKE 34,2: NORMAL

110 D$ =  CHR$ (13) +  CHR$ (4): ON
      PEEK (43239) <  > 70 GOSUB
     50000
120  PRINT D$"IN#0": PRINT D$"PR#
     0"
130 MODEMSLOT = 2
140  ONERR  GOTO 60000
150  GOTO 300: REM

200  FOR I = 1 TO 3: PRINT  CHR$
     (7): NEXT : RETURN
210  HTAB 8: INVERSE : PRINT "H";
     : NORMAL : PRINT "ALF": RETURN

220  HTAB 8: INVERSE : PRINT "F";
     : NORMAL : PRINT "ULL": RETURN

230  HTAB 8: INVERSE : PRINT "E";
     : NORMAL : PRINT "VEN": RETURN

240  HTAB 8: INVERSE : PRINT "O";
     : NORMAL : PRINT "DD ": RETURN

250  HTAB 8: INVERSE : PRINT "N";
     : NORMAL : PRINT "ONE": RETURN
     : REM

300  HOME : VTAB 3: PRINT "DUPLEX
     -";: GOSUB 210
310  VTAB 5: PRINT "PARITY-";: GOSUB
     230
320  VTAB 7: PRINT "DATA WORD LEN
     GTH-";: INVERSE : PRINT "7":
      NORMAL
330  VTAB 9: PRINT "STOP BIT-";::
      INVERSE : PRINT "1": NORMAL
     : REM

500  VTAB 3: HTAB 8: GET SETUP$: IF
     SETUP$ <  > "F" AND SETUP$ <
      > "H" AND SETUP$ <  >  CHR$
     (13) THEN  GOSUB 200: GOTO 5
     00
510  IF SETUP$ =  CHR$ (13) THEN
     DUPLEX$ =  CHR$ (8): GOSUB 2
     10
520  IF SETUP$ = "H" THEN DUPLEX$
      =  CHR$ (8): GOSUB 210
530  IF SETUP$ = "F" THEN DUPLEX$
      =  CHR$ (6): VTAB 3: GOSUB
     220
540  VTAB 5: HTAB 8: GET SETUP$: IF
     SETUP$ <  >  CHR$ (13) AND S
     ETUP$ <  > "E" AND SETUP$ <
      > "O" AND SETUP$ <  > "N" THEN
      GOSUB 200: GOTO 540
550 PARITY$ = SETUP$: IF PARITY$ =
     "E" OR PARITY$ =  CHR$ (13) THEN
      GOSUB 230
560  IF PARITY$ = "O" THEN  GOSUB
     240
570  IF PARITY$ = "N" THEN  GOSUB
     250
580  VTAB 7: HTAB 18: GET SETUP$:
      IF SETUP$ =  CHR$ (13) THEN
     SETUP$ = "7"
590 WRDLNGTH =  VAL (SETUP$): IF
     WRDLNGTH < 7 OR WRDLNGTH > 8
      THEN  GOSUB 200: GOTO 580
600  VTAB 7: HTAB 18: INVERSE : PRINT
     WRDLNGTH: NORMAL
610  VTAB 9: HTAB 10: GET SETUP$:
      IF SETUP$ =  CHR$ (13) THEN
     SETUP$ = "1"
```

Briefly, the operation of the program is:

- Check to see if FILEDUMP is in memory, if not, place there with the Lam routine. (lines 110 & 50000)
- Establish Micromodem and transmission protocols for parity, etc. Pressing [return] in response to input requests offers the following defaults:
  - Parity — EVEN
  - Stop bits — 1
  - Data word length — 7
  - Duplex — HALF
- Lines 200-330 format the screen, and lines 500-640 handle the input and establish the parameters.
- The value for the modem status word is established in lines 1000–2040
- Line 2500 checks whether a "table" (more about this later) needs to be sent or not.
- The user now has an option of sending a single file or a group of files. On the single file option, pressing [return] will print a disk catalog. A catalog is automatically printed if multiple files are selected.

Operation is as near automatic as possible. In the case of multiple files, any number may be specified, up to the capacity of one diskette, and the file names entered at one time. This then frees the user who may consume several cups of coffee while the files are being transmitted. In the event of a wrong file name or type, the error handler will skip that file and when transmission has been completed, print a list of any files requested but not sent.

In our specific useage, a "translation table" is sent prior to any actual text. This is actually a program that runs on the typesetting machine, and converts characters on the Apple keyboard to those on the typesetting machine. It also allows for the inclusion in text of special characters such as subscripts, copyright marks, etc. The PEEK in line 2500 checks to see if a table (or preliminary file) needs to be sent. If not desired, line 2500 may be changed to GOTO 3070. The name of the initial file is established in line 4000 as "T.TABLE." If this feature is used, the name may be changed to any appropriate title. To aid in understanding the program, long meaningful variable names have been used. They may be shortened if desired.

Here is a quite practical application program that can be easily modified to suit individual needs. If you have found it of use, why not drop a brief note to *Write -A.P.P.L.E.* and let others know how you have put it to work.

```
620 STPBIT =  VAL (SETUP$): IF ST
    PBIT < 1 OR STPBIT > 2 THEN
     GOSUB 200: GOTO 610
630  VTAB 9: HTAB 10: INVERSE : PRINT
    STPBIT: NORMAL
640  VTAB 16: PRINT "IS THE ABOVE
     CORRECT (Y/N) ";: GET SETUP
    $: PRINT SETUP$: IF SETUP$ <
     > "Y" AND SETUP$ <  > "y" THEN
     GOTO 300: REM

1000 FLAGSTTUS = 0: ON WRDLNGTH =
    8 GOTO 2000: ON STPBIT = 1 GOTO
    1020: ON PARITY$ <  > "E" GOTO
    1010:FLAGSTTUS = 1: GOTO 300
    0
1010 FLAGSTTUS = 5: IF PARITY$ =
    "N" THEN  GOSUB 200: GOTO 30
    0
1020 FLAGSTTUS = 9: IF PARITY$ =
    "N" THEN  GOSUB 200: GOTO 30
    0
1030  IF PARITY$ = "O" THEN FLAGS
    TTUS = 13
1040  GOTO 2500: REM

2000  IF STPBIT = 2 AND PARITY$ =
    "N" THEN FLAGSTTUS = 17: GOTO
    3000
2010  IF STPBIT = 2 AND PARITY$ <
     > "N" THEN  GOSUB 200: GOTO
    300
2020  IF PARITY$ = "N" THEN FLAGS
    TTUS = 21
2030  IF PARITY$ = "E" THEN FLAGS
    TTUS = 25
2040  IF PARITY$ = "O" THEN FLAGS
    TTUS = 29: REM

2500  ON  NOT  PEEK (0) GOTO 3070
    : GOSUB 4000
2510  GOTO 3070: REM

3000  PRINT D$"IN#"MODEMSLOT: PRINT
    D$"PR#"MODEMSLOT:SETUP$ =  CHR$
    (1):H$ =  CHR$ (8): PRINT SE
    TUP$DUPLEX$
3010  POKE 2040 + MODEMSLOT,FLAGS
    TTUS
3020  FOR I = 1 TO 500: NEXT I
3030  IF MULTFILES THEN  RETURN
3040  PRINT D$"FILEDUMP ";FYLNAME
    $
3050  PRINT D$"PR#0": PRINT D$"IN
    #0"
3060  IF  NOT (SGLFILE) THEN  RETURN

3070  HOME : VTAB 11
3080  IF  NOT  LEN (MULTFILES$) THEN
     PRINT "SEND MULTIPLE FILES
    (Y/N) ? ";: GET MULTFILES$: PRINT
    MULTFILES$
3090  IF  NOT (SGLFILE) THEN  INVERSE
    : PRINT "PLACE DATA DISK IN
    DRIVE, HIT ANY KEY";: NORMAL
    : CALL  - 756: PRINT
3100  IF MULTFILES$ = "Y" THEN 50
    00: REM

3200 SGLFILE = 1
3210 FYLNAME$ = "": INPUT "NAME O
    F FILE TO SEND -->";FYLNAME$
    :TB = 0: ON  LEN (FYLNAME$) >
    0 GOTO 3000
3220  PRINT D$"CATALOG": GOTO 321
    0
3230  END : REM
```

TELECOMMUNICATING WITH FILEDUMP

```
4000  POKE 0,0:FYLNAME$ = "T.TABL
      E": GOSUB 3000:FYLNAME$ = ""
      : RETURN : REM

5000  PRINT D$"CATALOG": INPUT "H
      OW MANY FILES TO SEND? ";MUL
      TFILES$:MULTFILES =  VAL (MU
      LTFILES$): ON  NOT MULTFILES
       GOTO 5000
5010  DIM FYLNAME$(MULTFILES),UNS
      ENT$(MULTFILES)
5020  FOR J = 1 TO MULTFILES
5030  PRINT "ENTER FILENAME "J;"
      ";: INPUT FYLNAME$(J)
5040  NEXT J: REM

5100  GOSUB 3000
5110  FOR J = 1 TO MULTFILES
5120  PRINT D$"FILEDUMP";FYLNAME$
      (J)
5130  NEXT J
5140  PRINT D$"IN#0": PRINT D$"PR
      #0": TEXT : HOME : IF (UNSEN
      TFLAG) THEN  FOR I = 1 TO MU
      LTFILES: GOSUB 6000: NEXT I
5150  END : REM

6000  IF  LEN (UNSENT$(I)) THEN  PRINT
      "FILE ";UNSENT$(I);" WAS NOT
       TRANSMITTED"
6010  RETURN : REM
```

```
50000 HEX$ = "BCDF:20 8E FD 20 A3
       A2 20 8C A6 F0 12 20 ED FD
      AD 00 C0 10 F3 8D 10 C0 20 1
      B FD C9 91 D0 E9 20 FC A2 60
       N A8E7:46 49 4C 45 44 55 4D
       D0 N 9D48:DE BC N A933:20 3
      0 N D9C6G"
50010  FOR I = 1 TO  LEN (HEX$): POKE
      511 + I, ASC ( MID$ (HEX$,I,
      1)) + 128: NEXT : POKE 72,0:
       CALL  - 144: RETURN : REM

60000  REM  *** ERROR HANDLER ***

60010 ER =  PEEK (222):EL =  PEEK
      (218) +  PEEK (219) * 256
60020  IF ER = 255 THEN  TEXT : HOME
      : END
60030  IF ER = 6 OR ER = 13 THEN
       PRINT "FILE "FYLNAME$;FYLNA
      ME$(J)" IS NOT ON THIS DISK.
      ": ON  NOT MULTFILES GOTO 32
      10
60040  IF ER = 6 OR ER = 13 THEN
      UNSENTFLAG = 1:UNSENT$(J) =
      FYLNAME$(J): ON J )  = MULTF
      ILES GOTO 5140:J = J + 1: RESUME

60050  PRINT "ERROR "ER" IN LINE
      "EL: END
```


DETACH YOUR KEYBOARD!

The KEYTEC Detachable Keyboard makes your Apple II perfect.

- 5 feet of keyboard freedom
- KEYTEC utilizes existing keyboard
- Easy 10 minute installation—complete instructions
- Apple warranty remains valid
- KEYTEC supports II+ shift-key mod
- Metal case is color-matched to Apple
- Includes cable, hardware and panel to replace old keyboard

YOU DON'T NEED A NEW KEYBOARD FOR $250 OR MORE...
YOU NEED THE KEYTEC IIe OR II+ FOR ONLY

We ship immediately by return mail.
Specify type of keyboard you wish to have (II+ OR IIe).

KEY TEC **$59**

P.O. Box 722, Marblehead, MA 01945
Phone your order: (617) 292-6484

VISA or MASTERCARD welcome. $3 shipping. Massachusetts resident $2.95 sales tax.

Circle 18 on Reader Service Card

RS232 SERIAL

AT LAST an APPLE® serial card that does it all! Communicate with a Modem, Remote terminal or Teletype. Just flip a switch and have letter quality serial printer protocol. BEST of all, only $44.95/kit, $59.95/assm. (Add $2.00 for shipping) Included is a RS-232 type connector & a comprehensive instruction manual. We also include an extended 6 month warranty and a terminal emulation program. Most orders shipped 48 hours after receipt*.

B & D ELECTRONICS:
Keeping the cost of your Apple® Byte size.
Send Check, Money Order* or COD* to:

B & D ELECTRONICS
P.O. Box 3651
Lawrence, Ks. 66044
(913)-841-4182
Circle 5 on Reader Service Card

Circle 5 on Reader Service Card

ELECTRONIC MAIL:

ANOTHER USE FOR THE APPLE

Bruce Bourbonnais

IN case you haven't noticed, the U.S. Postal Service is now providing a new electronic mailing system (E-COM). Their new system can be used by large main frame computers at 2400 to 4800 baud and also with small computers at 300 or 1200 baud. As an Apple user for about three years and always looking for a new project, an article that appeared in a local paper caught my eye. Before I knew what was happening (14 days), we were certified as a carrier for E-COM and a hobby turned into a business (QWIK MAIL) which could be made as large as you would care to go. As a help to other small computer users, we felt an article about E-COM would be of interest to either a user who would like to use the E-COM system in his/her own business, or as a carrier to hire out to other businesses.

Using E-COM, you can dial one of the 25 U.S. Systems Serving Post Offices and transmit one or two pages of text. Your text is printed, enveloped, sorted and delivered as First Class mail within a period of one or two days depending upon the area of delivery. The cost is 26 cents for one page or 31 cents for two pages. The first page can have 41 lines of text and the second page can have 56 lines. Should you want to use nine lines per inch, the first page can have 61 lines of text and the second page can have 84 lines. This cost includes the stamp, paper, envelope and sorting. For a user in his/her own business, this is very inexpensive. As a carrier performing the service for someone else, you would charge a service fee for your labor and the use of your system.

Your system can be an Apple][+ (DOS) (CP/M) (Pascal) with a modem installed. QWIK MAIL uses ASCII Asynchronous mode at 300 baud with a DC Hayes Micromodem and Hayes Terminal Program. This setup allows us to transmit 400 addresses every 33 minutes. We use CP/M with Wordstar in document mode for the text part of the transmission and the non-document mode for the address file. If you are a programmer, the Postal System will supply you, free of charge, the documentation needed to prepare a log-on program to dial into their system. If you are not a programmer, there are programs on the market for all systems.

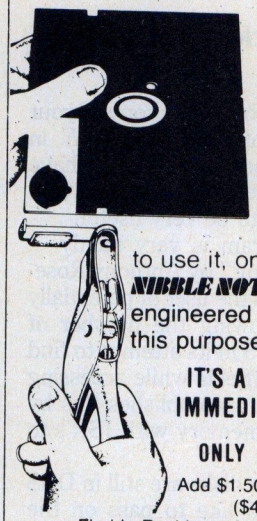
The varied applications for E-COM include invoices, new products information, price changes, newsletters, recalls, billing, election notices, etc. There are three message types accepted by E-COM. (SAM) Single Address Message sends one message to one address. This is not used very often. (COT) Common Text Message. This is used most often when you transmit a common text (1 or 2 pages) followed by your addresses. (TIM) Text Insertion Messages are a combination of (COT & SAM) and are used generally for invoices and statements.

E-COM is delivered in a very eye catching blue and white envelope which seems to motivate recipients to open this letter first. One of our first clients, for whom we transmitted a 1200 address application letter to their classes, advised us that they saved more than $500.00 over their previous mailing system plus hours of preparation time.

When we first decided to become certified, it seemed to be a complicated process. However, as we became further involved, we found the system to be quite easy along with the aid of the U.S. Postal Service. Simply get in touch with your U.S. Post Office E-COM Representative and let him/her coach you through the procedures. We feel you will find this a convenient and valuable service.

***DOUBLES DISKETTE STORAGE SPACE!!**

REDUCE YOUR DISKETTE COSTS BY 50%



*The back of your 5¼" single sided diskette has recording medium. All you need is an ACCURATELY placed ***"write enable notch"*** to use it, on many systems. ***NIBBLE NOTCH***™ is a precision engineered tool designed for this purpose.

IT'S A MONEY SAVER! IMMEDIATE SHIPMENT!

ONLY $14.95

Add $1.50 Postage/Handling ($4.50 Foreign)

— Florida Residents Add 5% Sales Tax —

ORDER TODAY!

SEND CHECK OR MONEY ORDER TO:

NIBBLE NOTCH ™

Division of Cortran International

4211 N.W. 75th Terrace, Dept. 1

Lauderhill, Florida 33319

PAT. PEND.

Circle 27 on Reader Service Card

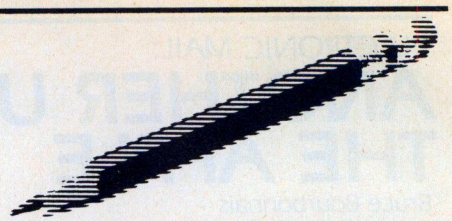
Computer Software Associates, an established software house, supplies software for APPLE, COMMODORE, and IBM home computers. CSA is currently looking to expand its product base of serious programs in the areas of home, education, and office.

Computer Software Associates offers over 30 years of market experience, worldwide distribution, innovative packaging, generous royalties, and programmer support.

Interested programmers should contact Mike Mahan, Product Development Department, Computer Software Associates, 50 Teed Drive, Randolph, Massachusetts 02368 Tel. (617) 961-5700.

COMPUTER SOFTWARE ASSOCIATES

Circle 12 on Reader Service Card



Error Checking in 3.2

Dear Editor,

I have recently obtained a copy of your latest volume of *Call -A.P.P.L.E.* in Depth, "All About DOS," and I have found the wealth of information contained in this book to be very useful.

One short program is very handy — "Disk Error Checking" by Anthony Rose. This program warns the user of potentially bad sectors by showing the number of retries made by DOS in its attempt to find the correct address fields while accessing data on the disk. It accomplishes this by modifying DOS in memory with JSR's to its routines.

Most of my older disks are still in DOS 3.2.1 format, so I'd like to pass on the appropriate locations to be used for this DOS version.

Line	DOS 3.3	DOS 3.2.1
16	STA $BDC9	STA $BDBC
18	STA $BDCA	STA $BDBD
20	STA $BDCB	STA $BDBE
22	STA $BE6E	STA $BE5F
24	STA $BE6F	STA $BE60
82	JMP $BE8E	JMP $BE7F

In DOS 3.2.1, if the number of retries exceeds $0D (13 sectors) you could be in trouble because DOS couldn't get the data the first time around. Note that the above locations will not work with DOS 3.2.0.

Incidentally, an extra space should be added to the header "TRACK" in line 36 of the assembly listing to correct a minor bug which causes a ")" to be displayed.

Richard J. Corcoran
433 Catherine St.
Somerville, NJ 08876

BSAVEing From Assembly

Dear Sir,

What I have to report is probably not new, but I didn't see it in *All About DOS* so perhaps it will be useful.

I am working on a system that requires binary data files to be prepared in BASIC, manipulated with a machine language call, and then printed out with BASIC. I needed a machine language routine to allow me to save a series of 16 binary files of 3K each. Reference books give details about using RWTS, but I couldn't divine how to find free sectors, add names to the directory, etc. I wanted something simple.

Using *Beneath Apple DOS,* I tried several different calls to DOS from my machine language program, but either DOS would perform a warmstart rather than return to my program or errors would yield the familiar ROGRAM TOO LARGE rather than return to BASIC for error processing. Eventually, I came up with the following which seems to work fine.

To perform a BSAVE from machine language:

Move an ASCII image of the command, exactly as you would type it, into the input buffer at $200. (Use hex as shown on page 7 of the *Apple Reference Manual*).

Store $01 in $AA52 (the "video intercept state")

Store in $AA5D an index to the position in the buffer of the carriage return ($00 represents address $200).

Load the Accumulator with $8D (carriage return)

JSR $9EBD (reference *Beneath Apple DOS* pages 8–6)

The command I put in the buffer at $200 was BSAVE SCORES A,A$8A00, L$C00 [CR]. I was able to modify the location of the title so that consecutive files had different names.

Thanks for a great magazine.

Art Armstrong
3345 Moore St.
Los Angeles, CA 90066

No N-Key Rollover

Editor:

Although I agree with Cecil Fretwell ("Comparing Apples and Apples," Volume VI, Number 6) that the Apple *IIe* is a great joy to use compared to previous Apple][keyboards (except that not being able to use right arrow to turn off escape mode is very annoying, as he pointed out), he is mistaken in saying the *IIe* keyboard has N-key rollover. Just type THE on the *IIe* (holding down all three keys) and you'll get something like THJE. Likewise, press all four arrow keys and an apostrophe incorrectly appears — very similar to his ASDF producing ASDF2 (on a][+) example.

You'd think Apple would be able to spring about a buck to buy the fifty-odd diodes needed to make the keyboard have N-key rollover! Unlike Cecil, I kind of like the raised bumps on the middle-finger keys. And the *IIe* keyboard is one of the few microcomputer keyboards with all ASCII characters — very nice. Bottom line: despite not having N-key rollover I find the *IIe* keyboard a big improvement over previous Apple][keyboards.

Philip Tubb
ALF Products Inc.
1315F Nelson St.
Denver, CO 80215

More Pascal //e

Dear Editor:

This letter concerns, and, to an extent, is in response to the article entitled "What in the Blaise is Going On?" in the June 1983 *Call -A.P.P.L.E.*

We have been working on an Apple *IIe* for about two months now, solely in Pascal. When we ran APPLE3:SETUP to produce a SYSTEM.MISCINFO designed specifically for the *IIe*, we made the following changes:

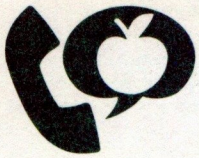
HAS LOWER CASE: now TRUE
KEY TO MOVE CURSOR DOWN: now [Ctrl-J]
KEY TO MOVE CURSOR UP: now [Ctrl-K]
(the two CTRL keys may also be entered in SETUP by just pressing the appropriate arrow key followed by [RETURN])

In our setup, we have the cursor respond to the four arrow keys and simply use the appropriate keys for the brackets ("[" and "]"). We have had no problem with using the "up-arrow" for cursor motion. Pascal responds to all of the arrows just as if they were ^0 and ^L on a][+, and we have had no problems with BIOS recognizing "up-arrow" as a "[".

It would seem that Pascal input filters through a routine that checks the character entered against the special characters that Pascal reads from SYSTEM.MISCINFO before bios gets hold of it.

We hope this information can be of help to *IIe* owners who program in Pascal. We think that the full keyboard of the *IIe* is a definite advantage to Pascal programmers and we are happy that it is so easy to take advantage of.

Jeff Gonion & David Bernholdt
523 Edgewood Lane
Elk Grove Vlg, IL 60007



MORE INFORMATION about the products and services in this issue FREE!

Just circle the number and fill out the card. (Note: card expires three months following month of publication.)

READER SERVICE CARD

Free Information on the Items circled

Card Expires November 30, 1983 (9-83)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210

1. What kind of product(s) are you likely to buy in the next three months?

A. ☐ Hard Disk
B. ☐ Increased Capacity Floppy Disk
C. ☐ Apple Compatible Floppy Disk
D. ☐ Computer
E. ☐ Printer
F. ☐ Modem
G. ☐ CRT

2. More information on Apple PugetSound Program Library Exchange? A. ☐ Yes B. ☐ No

Name _____

Company _____

Address _____ City _____

State _____ Zip _____ Phone _____

Act Now!

Join!

A.P.P.L.E.

Apple **P**ugetSound **P**rogram **L**ibrary **E**xchange

The **World's** Largest Apple Computer User Group

Be a part of the over 18,000 A.P.P.L.E. members and you'll receive:

- 12 issues of **Call —A.P.P.L.E.** magazine. The nation's premier Apple computer publication.
- Substantially reduced prices on sophisticated software, hardware, publications, and services.
- Access to the A.P.P.L.E. closed user group on the **SOURCE.** Excellent for information exchange and program downloading.
- **Call —A.P.P.L.E.** Hotline.

ALL THESE BENEFITS AND MORE FOR ONLY $45!

Detach and mail the Business Reply Application Form.

Call —A.P.P.L.E. Reader's Choice

Help us help you. By filling out and returning this card, marked with your preferences, you can guide us in selecting material tailored to your needs. Just place the numbers from 1 to 10, in order of preference, in the appropriate boxes. Please do not choose more than 10 features. A.P.P.L.E. members who include their membership number, will have their names placed in a drawing for $100 credit toward any A.P.P.L.E. product. Drawing will be held November 15, 1983. The top scoring author will also receive a $100 cash bonus.

☐ [1] **The Age of Silicon** Jonathan Post
☐ [2] **Where Has the Data Gone** Ken Kashmarek
☐ [3] **From BASIC to LOGO** Lee Reynolds
☐ [4] **CEEMAC** Jim Hockenhull
☐ [5] **The Graphics Magician** Harvey Gratt
☐ [6] **Applesoft Graphics Mover** Homer O. Porter
☐ [7] **Shape Table Splicer** Cyrus W. Roton
☐ [8] **Extra Memory for Spreadsheet** George Mochizuki
☐ [9] **Grade Book Programs** Wes Felty
☐ [10] **Rocky's Boots** Earle Hancock
☐ [11] **Magic Square Dance** J. Taylor
☐ [12] **Nellie and the Pirates** Rick Sutcliffe
☐ [13] **Ignoring Extra Ignored** Ari/Dov Freund
☐ [14] **A.P.P.L.E. Doctor** Bruce Field
☐ [15] **A Softcard LST Driver** Gerald Cahill
☐ [16] **And The Winner Is . . .** Ken Kashmarek
☐ [17] **A.P.P.L.E. Computer Camp '83** Charlie/Susan Stillman
☐ [18] **SYSTEM.LIBRARY** David Walker
☐ [19] **Pascal 1.2** Allen Munro
☐ [20] **ApMail Apple-cations** Ken Kashmarek
☐ [21] **The Gaming Room** Val J. Golding
☐ [22] **Inverse String Printer** Stowe Keller
☐ [23] **How To Become An Author** Daniel T. Lyons
☐ [24] **Telecommunications With Filedump** Steve Ross
☐ [25] **Another Use For the Apple** Bruce Bourbonnais
☐ [26] **Menu** Val J. Golding
☐ [27] **Write —A.P.P.L.E.** the readers
☐ [28] **Tomorrow's Apples Today** Kevin O. Donohoe
☐ [29] **Editor Bytes Back** The Editor
☐ [30] **Bit Converter** Val J. Golding
☐ [31] **So What Did You Expect?**

Name _____ Address _____ Memb. # _____

A.P.P.L.E. employees ineligible for prizes. Void where prohibited by law.

☐ Please send me an A.P.P.L.E. Apple-cation and information.
☐ $45.00 payment enclosed.
☐ Mastercharge/Visa

$25.00 First time fee
20.00 First year dues
$45.00 Total dues

MasterCard # _____ Exp. Date _____

VISA # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____

Zip _____ Phone _____

A.P.P.L.E., 21246 68th Ave. S., Kent, WA 98032
(206) 872-2245

First Class mail for **Call —A.P.P.L.E.** add $15. Otherwise third class mailed free as part of membership in U.S.A.

Additional Airmail postage required for foreign memberships and shipments.

Additional membership postage:
- Canada, West Indies, Central America, Columbia, Venezuela and Mexico — Add $15.00 ($60.00 total)
- All S. American countries (except above), Europe, and N. Africa — Add $21.00 ($66.00 total)
- Asia, Mid East, S. Africa, Australia and Pacific Islands — Add $27.00 ($72.00 total)

All orders must be prepaid in U.S. funds only.

Apple PugetSound Program Library Exchange
The WORLD'S largest Apple Computer user group

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 217 CLINTON, IA

POSTAGE WILL BE PAID BY ADDRESSEE

Call —A.P.P.L.E.
P.O. Box 1993
Clinton, IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA

POSTAGE WILL BE PAID BY ADDRESSEE

Call —A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA

POSTAGE WILL BE PAID BY ADDRESSEE

A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

Word processing.

It's simple.

The typewriter was a wonder in its time. **Letter Perfect**™ is a wonder for today. Letter Perfect from LJK lets you use your computer to write letters and create reports and edit your work. Change a word. Move a line. Combine two paragraphs. You can do it all with Letter Perfect, and more. Much more.

When it comes to practical software for Ataris and Apples, Letter Perfect is simple to learn. And hard to beat. Ask your dealer for a demonstration, or write LJK for more information.

LJK

LJK, INC. • 7852 Big Bend Blvd., St. Louis, MO 63119 • (314) 962-1855

Circle 15 on Reader Service Card

WRITE A.P.P.L.E.

CP/Mystery

Dear Sirs,

I started playing (so says my wife) CP/M a few weeks ago. I used J.D. Dennon's book *CP/M Revealed* to get acquainted with it. After typing in the program RESTORE (page 146), I found out that it does not always work. It took me several hours to trace the reason for the trouble, but it is quite simple: Function 17 (Search for first) does not behave as described in the CP/M 2.0 Interface Guide.

Function 17 is handled by a routine starting at D318H (56K). Because of a JZ at D33FH, if the DR field of the FCB contains an ASCII question mark the search ends when the last active entry in the directory is reached.

Therefore, if one wants Function 17 to behave as described in the Interface Guide, the following patch can be used within a program:

```
MVI   A,0C3H   ; change JZ
STA   0D33FH   ; into JMP
```

(For 44K change D33F into A33F)

J. Drabbe
78 Ave. Huysmans
1050 Brussels
Belgium

Loses Magfiler Files

Dear Val:

First, I'd like to say that *Call –A.P.P.L.E.* is probably the best thing that ever happened to my Old Apple][+. Please keep up the excellent work.

I'm writing to point out what might be considered a "quirk" of Applesoft BASIC and to show how a lack of error-checking in a program can cause disasterous results.

Normally, any non-numeric input to a statement such as

```
100 INPUT X
```

will result in a ?REENTER response. However, some programmers may not realize that if the user enters a space followed by a return, no error response will be given and the variable will be assigned the value 0.

This "quirk" can result in problems or even disaster in a program such as Jack Olivieri's MAGFILER (May '83). While this is a good program, it fails to check the user's input of a menu selection. A [space] [return] response to "Choose one" (line 230) will assign a 0 to the variable which is used in the ON...GOTO statement that follows. The results in the first statement number specified being executed. In the case of this program, execution goes to the "Create New Files Module" which clears all of the data files and *poof!* all of the existing data is lost!

A simple solution to this problem is to insert the statement

```
235 IF V5 < 0 OR
    V5 > 8 GOTO 230
```

which causes a loop back to the input statement if any invalid responses are given.

A little error checking can go a long way to prevent much screaming and gnashing of teeth!

A couple of other notes on MAGFILER. Line 1285 and Line 1286 appear to be reversed — the subroutine RETURNs before CLOSEing the files. Also a CLOSE statement appears to be needed at the end of the "Sort Module." To do so, make the following changes:

```
1460 NEXT J: NEXT I
1462 PRINT D$;"CLOSE"
1464 GOSUB 2: GOTO 170
```

As I said earlier, *Call –A.P.P.L.E.* is great and your staff and the program contributors do a great job!

Luke H. "Pat" Montgomery, III
2301 Lowell Drive
Lawrence, KS 66044

PIE Writer User Group

Dear Editor:

Please consider publishing the following open letter to all PIE users:

If you are an Apple PIE or PIE Writer word processor user with an Apple][or *IIe*, then you will be interested in a new Users Groups now forming:

Apple PIE Writers

As the name indicates, we plan to support all Apple versions of PIE. This support will include tips, tricks, modifications, help, and just about anything to enhance your use of the already outstanding PIE word processor.

If you have made some modifications to PIE let us know; maybe others would be interested. If you have questions ask the Group; somebody will have the answer. This is your Users Group, we are just the clearinghouse.

Depending on the response we receive, a newsletter will be published whenever we have sufficient material to warrant one.

Right now membership is free, so if you are a PIE user, send us your name, Source or Compuserve ID (if applicable), and your US Mail address. Include which version of PIE you use: Apple PIE or PIE Writer, 40 or 80 column (specify your board), and][or *IIe*.

Send to: Apple PIE Writers
12841 Hawthorne Blvd,
Box 589
Hawthorne, CA 90250
or Source: CL1312
Compuserve: 74405,764

Monty Lee
Mike Weasner

Tricking DOS

Dear *Call –A.P.P.L.E.*

Regarding Dave Garson's letter (May *Call –A.P.P.L.E.*)

Regarding Ralph Swerdlow's "Controlling Control D"

I, too, have had problems using the CHR$(13) + CHR$(4) technique.

DOS "watches" the characters being output from programs running in the Apple waiting for a [cr] followed by a [Ctrl-D] and then parses the entire line of output in search of a DOS command. DOS keeps track of what is happening in the process by means of a state flag referred to as "CSWSTATE." The value DOS stores in CSWSTATE is determined by the character being output and the value in CSWSTATE determines how the next character to be output will be handled. Basically the CSWSTATE flag is a clever way of keeping track of the last character output and a way of controlling how the next character output will be handled the next time we pass through this particular bit of code.

The CSWSTATE is stored at $AA52 (43602).

By POKEing 43602,5 you can "fool" DOS into thinking that the last character output was a [cr]. You can test this technique by means of the following test program:

```
10 PRINT " ";
30 PRINT CHR$(4)"CATALOG"
```

When this program is run instead of a catalog you'll get —

```
CATALOG    ([space]CATALOG)
```

Now insert line 20

```
20 POKE 43602,5
```

When the program is run again, DOS will notice the CATALOG command because line 20 told DOS that the last character output was a [cr] rather than the [space] that was really the last character DOS saw output from the test program.

John Korntved
4773 Snow Drive
San Jose, CA 95111

TOMORROW'S APPLES TODAY

Kevin O. Donohoe/editor

NOTE: These products are presumed to be state-of-the-art. Call -A.P.P.L.E. does not necessarily endorse products published in this column as we have not tested them.

FANTASY PLAZA AN ELECTRONIC SHOPPING FANTASY

Fantasy Plaza is a revolutionary concept that simulates a shopping center from within a computer. Anybody having access to a computer terminal and modem can experience the fun of shopping regardless of their location. The Fantasy Plaza computerized shopping mall is accessible by modem at 1-213-244-1100.

For those not lucky enough to own a modem, Fantasy Plaza also provides a toll-free telephone ordering, and mail order using our unique and humorous catalog that has been designed to resemble a Plaza Directory. Customers within the continental United States can order by calling 1-800-824-7888, or for Alaska and Hawaii, 1-800-824-7919 and ask for our Plaza Operator #12. By mail, contact Fantasy Plaza at P.O. Box 6055, Burbank, California 91510. Additional information can be obtained at 1-213-840-8211.

Customers shopping by computer or browsing the catalog will notice a unique selection of merchandise. Fantasy Plaza carries software and hardware products for most of today's most popular personal and home computers. These products occupy floors 2 through 5 in the Plaza.

After the customer is connected into the Plaza, he or she is transported to the proper floor by a beautiful imaginary glass elevator. A map of the floor is then displayed and, using simple commands, the customer can "walk" into any desired store. Customers may encounter other people along the way, and may have the opportunity to "chit-chat" with another Fantasy Plaza customer connected on one of the other incoming computer lines.

Once the customer has completed selecting the items he or she wishes to purchase, the customer returns to the "lobby" on floor 1 and chooses the method of payment with the Plaza Cashier. Fantasy Plaza accepts VISA and MASTERCARD, or the merchandise can be shipped COD. Or, Fantasy Plaza can send the customer a list of the items, and process the order when the payment is received. The customer will give the Plaza Cashier other required ordering information. The customer is then escorted to a waiting limousine for the return home from this electronic shopping experience.

Fantasy Plaza is a partnership between Andres Munoz of Munoz Enterprises, and Gregg Collins of Matchmaker Enterprises, the company that developed Dial-Your-Match, the first free home computer accessible bulletin board dating service, accessible by modem at 1-213-842-3322. A list of the other Dial-Your-Match computer systems is also available by calling that number.

Fantasy Plaza, P.O. Box 6055, Burbank, California 91510, (213) 840-8211.

APPLE SPONSORS STUDENT COMPETITION

Apple Computer, Inc. is offering kits for organizing student computer clubs to 10,000 elementary and secondary schools nationwide. An Apple-sponsored national student competition also is underway in which both computer clubs and individual elementary and secondary school students will compete for travel, cash, and computer equipment prizes worth over $100,000.

Clubs and individual students that enter the competition are encouraged to submit projects in which they use microcomputers to serve the community. Independent judges, selected for their prominence in the computer and education fields, will judge the competition entries for their overall excellence. Competition finalists will be flown to Washington, D.C. in May, 1984 for a five-day stay during which the winners and runners-up will be named.

How to Obtain a Computer Club Kit

The kits are designed for schools with or without computer equipment. Apple will send the kits to the first 10,000 elementary or secondary, public or private schools which request them. The letter must be on school stationary, be signed by the school principal and include the name of an adult sponsor either a teacher, administrator or parent—willing to advise a club that has a minimum of 12 interested students.

National Competition Details and Awards

The national competition offers student computer clubs and individuals an opportunity to show excellence and be recognized for their achievements. Although service-related projects are encouraged, any activity which uses a microcomputer will be considered for an award.

The winning club in each category will receive ten Apple *IIe* "Starter System" packages which include an Apple *IIe* personal computer with 64K of internal memory, one floppy disk drive with a controller card, a display monitor, and the Apple 80-column card.

In each category, there will be four finalists and 25 semi-finalists. One winner and three runners-up from each category will be selected in Washington D.C. The winners will each receive $1,000, the runners-up will each receive $500, and the semi-finalists will each receive $100.

Both club and individual entries in the national competition are due March 1, 1984 and should be mailed to the address below. Semi-finalists will be announced on April 4, 1984. Finalists will be announced April 18, 1984. The trip to Washington D.C. and selection of the winners and runners-up will be the week of May 16, 1984.

Kit Availability

The computer club kits will be available in both elementary and secondary versions this September.

Schools interested in receiving a kit or more information about the competition can call (617) 452-9979 or write to: Apple Computer Clubs, Box 948, Lowell, MA 01853.

TEACHER'S APPLE

Teacher's Apple is a grade tracking program for teachers. Teacher's Apple allows a teacher to track up to 50 students with up to 200 assignments a student. An assignment consists of an assignment name, assignment number, points possible, points received, and the date of the assignment. Teacher's Apple will print the assignments in many formats; highest to lowest grade, grade book column format, just percentages, any range of assignments, and complete printouts. As printer (to take full advantage of Teacher's Apple a80 column printer is recommended). Teacher's Apple also includes an editor to change data that has been entered incorrectly. $34.95. Simple Software Systems, P.O. Box 41069, San Jose, CA 95160.

NIBBLE EXPRESS III
YOUR APPLE* WILL LOVE IT!

TABLE OF CONTENTS

Turtle Graphics	8
Applesoft Line Cruncher	14
TRAC Income System	19
MAMA	25
Amper Free Space Catalog	26
Machine Language Editor	27
Apple CALENDAR	36
Peaceful Coexistence	43
Apple Slugger	43
Reverse Key	45
Amper Find	45
Bond Manager	47
DOS 3 + 2	53
Amper Speed	57
Apple Darts	61
MicroCalc	63
Variable Cruncher	72
Life	77
Apple Flash	80
Checker	81
GO- Greeting Program	83
Quick Sort	86
Applesoft Line Editor	90
Othello	96
SPRINT	99
Disk Dump	102
Hi-Res Colors	103
Apple Record Command System	104
Apple Art Gallery	114
Game I/O Expmt	119
Super Keypad	123
Disk Commander	125
Compare Applesoft	132
Amp-L-Soft	136
Electronic Message Center	152
The Shape	155
Disk Map	159
Apple Bowl Football	161
Apple Scroller	167
Lower Case Letters	168
Recipe Box	170
Disk Doctor	178
Quasar II	182
DOS Command Entry	189
MLE Driver	189

NOW AVAILABLE!

NIBBLE EXPRESS is an Anthology of the Major Articles and Programs appearing in Volume 3 of NIBBLE Magazine . . . nearly $400 worth of Programs for your Apple II, ACE, or other Applesoft-compatible computer!

NIBBLE EXPRESS III contains *up-to-date enhancements, KeyPerfect Tables,* and the *Best of NIBBLE!*

Even if you have all 8 issues of NIBBLE in 1982, you'll want the Express to have updated program listings in one convenient package! It's a MUST for your Library!

NIBBLE EXPRESS III is a bargain at $17.95 plus $1.75 Postage/Handling ($2.50 outside the U.S.).

YOUR APPLE WILL LOVE IT!
(And so will you!)

NIBBLE
P.O. Box 325
Lincoln, MA 01773

Yes! I want to reserve NIBBLE EXPRESS Vol. III for my Library! Here's my ☐ Check ☐ Money Order for $17.95 plus $1.75 postage/handling. (Outside U.S. add $2.75 for postage/handling).

☐ Also send me NIBBLE EXPRESS Vol. I at $12.95 Plus postage/handling (see above).

☐ Send me NIBBLE EXPRESS VOL. II at $14.95 plus postage/handling (see above).

Master Card & Visa Accepted

Card # _____ Expires: _____

Signature: _____

Telephone: _____

Please Print Clearly

Name: _____

Address: _____

City _____ State _____ Zip _____

Your check or money order must accompany your order to qualify.
Outside U.S.: Checks must be drawn on a U.S. Bank.

*Apple is a registered trademark of Apple Computer Inc.
*ACE is a trademark of Franklin Computer Co.

Circle 26 on Reader Service Card

TOMORROW'S APPLES TODAY

THE TEACHER'S AID

The best in classroom management systems. A complete system for class lists, attendance, grades, comments, and reports. Designed for individual junior/senior high school teachers. Developed under the direction of high school teachers. Requires a 64K Apple][or a 48K Apple][Plus and two disk drives. Guaranteed error free. For more information, send $1 to: Seattle Software, P.O. Box 31943, Seattle, WA 98103, or send $59.95 check or money order to receive an immediate copy, complete with users manual.

GREAT ENTERTAINMENT AT LOWER PRICES

Insoft, who together with Paul Lutus brought you GraFORTH and Electric Duet, has announced new low prices on its Apple Arcade game series. The suggested retail price of Grapple, Spider Raid and ZARGS — all written in GraFORTH — is now only $19.95, a price reduction of 33 to 43%.

"The Apple entertainment software market is extremely competitive" according to Insoft's president, Michael Brown. "Our products have always represented exceptional value and the new pricing structure reflects our continued commitment to that goal."

According to Insoft, several distributors and retail chains are looking closely at the product line due to its competitive pricing structure and attractive packaging. "The games have sold well at $29.95 and $34.95 but we expect our sales to take a quantum leap at the lower price," according to Mr. Brown.

Insoft, Inc., P.O. Box 19208, Portland, OR 97219, (503) 244-4181.

AIRSIM-3 AIRPLANE SIMULATOR

This is a flight simulator which represents a typical private plane with all the instrumentation required for instrument flying. There is a standard instrument panel. In addition, there is an Automatic Radio Direction Finder (ADF), a VOR display, and a dual VOR/HSI display with glide slope indicator which can be used for ILS approaches. There are numerous navigation aids in the program data base, so that long-distance IFR flights can be made. There is Distance Measuring Equipment (DME) and OM and MM indicators. The user can create custom landing problems, including Nav-Aid locations, runway location and direction, and wind conditions, including a graphics display of the runway.

AIRSIM-3 is aerobatic, being able to perform loops, rolls, stalls, and other similar maneuvers. There is scenery along the California coast from Los Angeles to San Francisco, with numerous fields equipped for instrument approaches.

AIRSIM-3 has been designed to be a real aid for beginning and accomplished pilots. Nonpilots will enjoy its many features, and will find AIRSIM-3 a pleasant and simple introduction into the many complexities of airplane operation.

AIRSIM-3 will run on an Apple][,][+, and *//e* computers. It is available from Mind Systems Corporation, P.O. Box 506, Northampton, MA 01061. Phone: (413) 586-6463. Price: $40.00.

NEW PHYSICS LAB INTERFACES FOR APPLE COMPUTERS

Cross Educational Software, a publisher of educational software for high school and college science courses, announces three new lab interfaces for Apple computers — HEAT, LIGHT and SOUND. These interfaces can be used for classroom physics experiments. Each package contains a kit of parts that can be plugged into the game connector, a disk for calibration, and documentation for several experiments.

LIGHT experiments include timing a pendulum, measuring the acceleration of gravity, measuring light intensity, and the efficiency of a light bulb. Four phototransistors are included. $60.

HEAT experiments include graphing temperature vs time, thermal radiation, cooling curves, and specific heat. Four thermistors are included. $60.

SOUND experiments include sound intensity, simulated oscilloscope, and Fourier spectrum analyzer. A speaker, microphone, potentionmeter, transistors and a capacitor are included. $60.

Cross Educational Software produces a series of 114 physics programs for Apple computers. 70,000 of these programs are now being used in over 2,000 schools. There are 12 disks in the series with 6 or more programs per disk. Prices range from $15 - $35 per disk with the entire set costing $260.

For more information contact Cross Educational Software, P.O. Box 1536, Ruston, LA 71270, (318) 255-8921.

BUS RIDER

The BUS RIDER logic analyzer, by R.C. Electronics, Inc., is an essential tool for developing hardware boards, interface software, or debugging assembly code for the Apple computer. The BUS RIDER features real time monitoring of the Apple][and up to four external signals ranging from -12V to +12V. Operational software provided on a floppy disk loads into the Apple RAM for turnkey operation.

The BUS RIDER monitors all bus signals including 16 bit address, 8 bit data, IRQ, R/W, DMA, NMI and four external lines. Trigger options are provided on the BUS RIDER to enable the circuit to trigger on any combination of bus signals and external signals. In addition, separate, pre-trigger viewing of up to 510 samples is possible with the system. An operating speed of 1 Mhz allows analysis of 512 samples in either the interrupt mode, resulting in immediate display after acquiring 512 samples, or the passive mode, which displays after referral to the BUS RIDER software.

The BUS RIDER is available at a cost of $395.00 which includes the circuit card, reference manual, cables for external inputs, and software. For more information, contact R. C. Electronics, Inc. at 5386 Hollister Avenue, Suite D, Santa Barbara, CA 93111, or phone (805) 964-6708.

SPEED READER II

Davidson & Associates announces its newest product, SPEED READER II, a complete reading improvement program for high school, college, and adult students, was designed by reading specialists.

SPEED READER II offers six stimulating activities to help the user increase reading speed and build comprehension.

A Timed Reading Test electronically computes the user's reading speed in words per minute and then tests comprehension. Warm-up exercises strengthen eye muscles to enlarge eye span and sharpen perception. Lessons in eye movement, column reading, and text reading tran the user to become a more efficient, more effective, more productive reader. These activities access a data disk, which contains 35 reading selections.

Also included are an editor which allows the user to enter additional reading material and a grade level analyzer which measures the reading level of the material entered.

SPEED READER II is an updated and expanded version of SPEED READER, published by Apple and Davidson & Associates last year.

The two-disk package contains a 60 page instructional manual with a course outline. The suggested retail price is $69.95.

Eric Gaer Company, 22420 Sentar Road, Woodland Hills, CA 91364, (213) 704-0683.

TOMORROW'S APPLES TODAY

MICRO-BASED BILLING SYSTEM FOR DOCTOR'S

High Technology Software Products, Inc. announces the availability of the Doctor's Office Companion™ in August of 1983 for CP/M based Hard disk computer systems. D.O.C. has been available for Apple][computers with Corona or Corvus hard disk drives since March of 1982.

Patient data is gathered for automatic printing of the HCFA 1500 Universal insurance form, as well as forms required by some of the states which have not yet adopted the "universal" form.

D.O.C. gives particular attention to the insurance billing activities, but also provides other payment types, such as cash payment, among others. The office can define up to 15 income categories and up to 10 "street" accounts. Zero-balance billing is supported for those states in which it is required.

The CPT file is customizable by the user, so just the CPT codes and descriptions which apply to the physician's office need to be entered, and only one time. On daily patient transactions, the user simply enters the CPT code and D.O.C. prints the corresponding description, thus minimizing entry time and minimizing typographical error.

REPORTS GENERATED INCLUDE: guarantor list with or without balances; CPT list with codes, descriptions, and the price most often associated with the service; Transactions analyses, daily and monthly, giving the income totals by doctor and by income category, and total charges by service type; Monthly ageing report, History report of all transactions for any guarantor.

QUERY AVAILABLE: Current balance query gives current 30, 60, 90, 120+ and total balances.

Doctor's Office Companion is distributed worldwide by High Technology Software Products, Inc., 1611 N.W. 23rd Street, P.O. Box 60406, Oklahoma City, OK 73146. (405) 524-4359. Dealer inquiries welcome.

MONTE CARLO SIMULATIONS

Actuarial Micro Software announces MONTE CARLO SIMULATIONS (MCS), a unique utility which allows the user to prepare for many seemingly unpredictable occurrences (such as emergency health-care or automobile expenses). MCS features a non-technical approach, sophisticated graphic displays, High-Res shapes, color, and sound effects. You can see and hear the processes as they take place. Hundreds of years of experience can be simulated and analyzed in just a few seconds using this software.

MCS is fully functional as a stand-alone utility, but can also communicate with VisiCalc through DIF files. MCS can be used in a variety of applications including: personal financial planning, forecasting, risk management, social sciences, education, self funding, corporate financial planning, business, and research.

The STATISTICAL ANALYSIS process uses the Chi-Square Goodness of Fit Test to match a set of raw data to a standard probability distribution. A variety of other statistical information is also produced. The SIMULATION process generates random numbers based on an assumed probability distribution using a Monte Carlo method. The simulated values are then available for use in whatever statistical model the user wishes.

MCS is a high-quality statistical analysis and simulation system developed and supported by actuarial and computer professionals. Moreover, it is user-friendly and well-documented so that the user need not be a statistical or computer expert to benefit from all its features. MCS requires a 48K Apple and one disk drive (3.3). A second disk drive and a printer are optional. The software and manual are available for $60. A more technically-oriented version which includes the source code is available for $90.

Actuarial Micro Software, 3915 A Valley Court, Winston-Salem, NC 27106, (919) 765-5588.

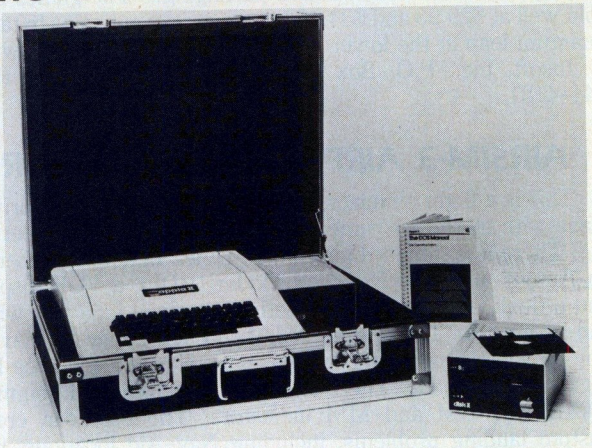
THE ORANGE INTERFACE. . .

Orange Micro announces the new Orange Interface for Apples and printers. This low cost parallel interface joins the #1 selling Grappler+ series and the Bufferboard to complete Orange Micro's line of Apple printer interface equipment. Priced at just $87, the Orange Interface has over 15 firmware commands for text screen dumping and formatting on Apple][,][+ and *//e* computers. And the Orange Interface is compatible with most all parallel printers.

Among the Orange Interface's innovative features is an 80 column screen dump for the Apple *//e*. Other commands include a 40 column screen dump, page length and margin sets, add or delete line feeds, and much more. When not in use for formatting, the Orange Interface acts as a standard parallel interface, compatible with virtually all Apple software, CP/M and PASCAL.

Priced below many plain interfaces, the orange Interface offers a lot more for the money. A suggested list of $87 includes complete documentation, a 5 foot parallel printer cable and a full 90 day warranty. The orange Interface is available now through Apple dealers and computer stores worldwide. For more information contact Orange Micro, 1400 N. Lakeview, Anaheim, CA 92807, (714) 779-2772.

PROTECTIVE CASES FOR HARDWARE



Cases, Incorporated, a 17 year old Seattle manufacturer of protective cases for various applications, is currently offering in-stock and custom manufactured protective carrying cases for computers and peripheral equipment. The firm offers two major types of hardware case — the Flight Form and C-Series — both of which withstand and isolate delicate computer hardware from the hazards of travel or shipping.

Both the Flight Form and C-Series cases features form fitted foam lined interiors, and track mounts are available to hold equipment securely in place. Both cases can also accommodate books, papers and software for the computer, and custom interior configurations can be specified. The firm offers a variety of hardware and accessories for each case; special clasps, strapping — even custom printing of loops — and a wide range of colors is available on custom orders. Standard order Flight Form cases are available in either white, black or royal blue, and standard C-Series cases in black or brown.

The company has an impressive array of cases for major hardware brands in stock: Apple as well as custom designed cases for electronic test equipment, medical instruments and photographic gear.

For further information contact Mr. Glenn Stenson, Cases, Inc., 13102 Stone Avenue North, Seattle, WA 98133, (206) 365-5210.

CONSULTANTS

John A. Hall, Jr.	Apple //e (Eastern U.S.)	(617) 241-9050
Cecil Fretwell	Apple //e (MidWestern U.S.)	(319) 236-0961
Bob Huelsdonk	Apple /// (Western U.S.)	(206) 362-4910
Harry T. Hanson	Apple /// (Eastern U.S.)	(201) 467-0712
Carmen Picardi	Apple /// (Eastern U.S.)	(603) 895-4447
Stuart Yusem	Apple /// (Eastern U.S.)	(609) 822-9306
Tim Smith	Apple /// (Midwestern U.S.)	(389) 734-7451
Neil Clemmons	Agriculture	(314) 968-9618
Erv Edge	Assembly (Alaska)	(907) 337-7620
Dennis Baumgarte	Assembly (Eastern U.S.)	(914) 298-8756
Michael Thyng	Business	(206) 524-2744
Murray Arnow	CPM/Engineering Applications	(312) 291-9210
Steven E. Blair	Data Bases	(913) 782-5901
Jerry Rogan	Data Bases, Business	(319) 332-4166
William P. Steinberg	DOS (Eastern U.S.)	(212) 921-0663
Art Schumer	DOS (Northern U.S.)	(701) 282-7907
Lee Meador	DOS (Southern U.S.)	(817) 469-6019
Kurt Thams	DOS/Gen'l (Western U.S.)	(415) 327-2116
WM. Reynolds III	DOS/Tech (Western U.S.)	(503) 472-4028
Ted Perry	Education	(916) 485-1690
Cornelis Bongers	European Consultant	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Jim Eatherly	Games	(202) 232-6046
Tedd Sperling	Geoscience	(517) 393-1296
Stan Axelrod	Hardware Design/I-O Control	(415) 563-1840
Tom Deal, M.D.	Handicapped Applications	(206) 634-2847
Randy Halvorson	Hardware Interfacing	(206) 783-9040
Ed So	Hi-Res	(408) 267-3448
Stowe Keller	Mathematics	(804) 973-7647
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Stephen F. Lee	Music	(704) 289-4102
Tom Woteki	Pascal (Eastern U.S.)	(202) 547-0984
Tim Smith	Pascal (Midwestern U.S.)	(309) 734-7451
Paul E. Sleeper	Pascal (Western U.S.)	(415) 494-1915
Bruce E. Sargent	Pascal (Western U.S.)	(206) 488-1161
Dennis R. Cohen	Pascal (Western U.S.)	(213) 956-8559
Earl Keyser	Pilot Language	(612) 429-5570
Randy Warrens	Pilot Language	(608) 837-9566
Hersch Pilloff	Stock Market	(301) 292-3100
Jim Hockenhull	Video Interfacing	(509) 334-4137
George L. Hefter	VisiCalc (East)	(617) 256-8148
Vince Aderente	VisiCalc (North)	(312) 289-9685
Dr. Bob Richard	VisiCalc (South)	(504) 466-0894
Brian Abrahams	VisiCalc (South Africa)	(Durban) 315442
Michael Ginsberg	Word Processing	(805) 688-1465

So What Did You Expect?
by Stowe Keller

```
10  PRINT "WHAT WAS
      ELLIOT'S FRIEND'S
      PROBLEM?"
20 L =  PEEK (105) +
      256 *  PEEK (106):
      FOR X = L - 12
      TO L - 4: POKE
      X, PEEK (X) +
      1: NEXT
30  LIST 100: END
100 DS AT  THEN  HTAB
```

```
5  REM
       SO WHAT DID
     YOU EXPECT
                 BY
     RALPH H. SWERDLOW,
    M.D.

          CALL- A.P.P.L.E


100 Z$ = "837932877265843268736832897985326980880
    69678463"
110  HOME
120 V = 1
130  FOR N = 1 TO
     LEN (Z$) STEP
     2
140  VTAB V
150  PRINT  CHR$ (
     VAL ( MID$ (Z$,N,2)))
     ;
160  NEXT N
170  PRINT
180 V = V + 1
190  IF V ( 24 GOTO
     130
```

FOREIGN MEMBERS: Please calculate postage based on weights in product descriptions.
(See postage table below for cost)

FOREIGN SHIPPING CHART

POSTAL TABLE		I W. Indies C. America Columbia Venezuela	II S. America (except in Table I) Europe, N. Afr.	III Asia, Mid East, S. Africa, Australia Pacific Is.
0 oz	1 oz	$0.50	$0.60	$0.70
1	2	.76	.96	1.16
2	3	1.02	1.32	1.62
3	4	1.28	1.68	2.08
4	6	1.54	2.14	2.74
6	8	1.80	2.60	3.40
8	10	2.06	3.06	4.06
10	12	2.32	3.53	4.72
12	14	2.58	3.98	5.38
14	16	2.84	4.44	6.04
16	18	3.10	4.90	6.70
18	20	3.36	5.36	7.36
20	22	3.62	5.82	8.02
22	24	3.88	6.28	8.68
24	26	4.14	6.74	9.34
26	28	4.40	7.20	10.00
28	30	4.66	7.66	10.66
30	32	4.92	8.12	11.32
2 lbs	2.5	5.96	9.96	13.96
2.5	3.0	7.00	11.80	16.60
3.0	3.5	8.04	13.64	19.24
3.5	4.0	9.08	15.48	21.88
Each additional ½ pound over 4 pounds add:		1.04	1.84	2.64

CANADA SHIPPING RATES

Up to:	
1	.20
2 oz	.37
3 oz	.54
4 oz	.71
5 oz	.88
6 oz	1.05
7 oz	1.22
8 oz	1.39
9 oz	1.56
10 oz	1.73
11 oz	1.90
12 oz	2.07
1 lb	2.58
2 lbs	3.10
3 lbs	4.10
4 lbs	5.10
5 lbs	6.10
6 lbs	7.10
7 lbs	8.10
8 lbs	9.10
9 lbs	10.10
10 lbs	11.10

For weights over 10 lbs, add $1.00 per pound.

ORDER FORM

Membership # _____ Expiration date _____ Date _____

Name _____ Address _____

City _____ State/Country _____ Zip _____ Phone _____

PRODUCT	Unit Price	*FOREIGN SHIPPING WT.	AMOUNT
1			
2			
3			
4			
5			

Mail to: A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245
24 hour toll free order line
1-800-426-3667 except in WA.
Orders only **NO** information

VISA Account No. _____ Exp. Date _____

MasterCard Account No. _____ Exp. Date _____

Subtotal _____

**Washington sales tax _____

TOTAL AMOUNT _____

*Foreign Members Only
**Washington state residents add 7.9% sales tax.
Sorry, no foreign hardware orders.

UNCLASSIFIEDS — A.P.P.L.E. ORDERS

Call -A.P.P.L.E. (un)Classifieds

Members Non-commercial rate .05/word
Members commercial rate .25/word
All others: 1.00/word, $25 minimum
Send ad copy with payment to editorial office. Unclassified copy must be received by the first day of the month preceding the date of issue.

MICROMOVER transfers files from one Apple to another Apple over a phone line. Files may be transferred without first being converted to text files. Data compression algorithm cuts transfer time. Disk-to-disk transfer of one, some or all files on a disk with minimum operator intervention. Works with DOS 3.2 or 3.3 and Hayes Micromodem II, acoustic modems and other modem types. Only one copy of the program need be purchased as it transfers a copy of itself to the distant end, if required. An excellent manual is included. Cost is $50.00 from DAB Systems, 9116 Rockefeller Lane, Springfield, VA 22153.

APPLE PASCAL ANIMATION GRAPHICS Morwe/Graphics - 83 pp. manual and two diskettes. SASE for details. $3.95 for manual. **-$19.95-** for complete kit. HENDERSON ASSOCIATES, 980 Henderson Ave., Sunnyvale, CA 94086.

FRENCH AND MULTI-LINGUAL Software for the Apple] [+ or //e. FREE CATALOG. LE PROFESSEUR, P.O. Box 301A, Swanton, Vermont 05488 (514) 747-9130.

LOW APPLE SOFTWARE PRICES! Up to 30% off. Check out our fantastic prices. Write for our FREE price list! KERR SOFTWARE, Box 5301-CA, Long Beach, CA 90805. Or call (213) 428-8193.

WANT TO WRITE ON THE BACK of your disk? Use our self-aligning Notch-opunch to add a write-protect notch. Send check for $8.95 (ppd., U.S.A.) Geisler Enterprises, 20333 Old Town Rd., Star Rte. #2. Tehachapi, CA 93561.

NOT JUST A GRADE BOOK! A teacher tested, comprehensive grading and report system! Adapts to Elementary through College and/or Business School courses. Completely flexible — YOU specify # of grade periods and length, letter grade requirements, one or two semesters, & more. 4 screen & 4 printer reports (MX-80). 48K, DOS 3.3,] [+ or //e. FREE INFO & SAMPLE REPORTS. 23 page MANUAL $2 (REFUNDABLE). Program disk & Manual $35. C&H Systems, 7997 Megwood, Memphis, TN 38134.

1983 SOFTWARE MARKETING & DEVELOPER'S SOURCE-MANUAL. How to effectively market your computer programs. Avoid pitfalls that cost you money. Copyright procedures — Marketing on cassette a License requirements — Starting a software business — Software markets — MORE! $19.95 postpaid. Celestial Software, 3010 Warr Lakeland, FL 33803.

FOR SALE: DTACK GROUNDED 68000 BOARD for Apple, 92K RAM. 12 available. New. Paid over $1100. Best reasonable offer accepted Call (617) 456-8663.

ALMOST WHOLESALE — Save BIG BUCKS on almost all Hardware & Software needs. Usually 20% to 45% below retail. Monthly specials. Send self-addressed stamped envelope for latest price list, or contact us to request quotes on specific items. COMPUTER PRODUCTS UNLIMITED, Apple Dept., 11116 East 46th Ave., Spokane, WA 99206. (509) 922-1346 (eves). CompuServe (73075,613). SOURCE (TCG249).

VIDEOTERM OWNERS — New character sets now available: HI-LIGHT/LO-LIGHT (great with WordStar); APL; SQUARE; BOLD FACE; FOREIGN LANGUAGES as well. Only $19.95 each, hi-light/lo-light pair $34.95. (Plus 7.6% tax if Washington resident). Add $1.00 for shipping/handling. Custom character sets available on a quotation basis. Write for literature including pictures of character sets. KJT Enterprises, P.O. Box 762, Mercer Island, WA 98040.

SYNETIX SOLID STATE DISK, 294K Double Drive with DOS and PASCAL software. Cost $530. Used one month and in perfect condition, $475, postpaid. Bailey, 115 Van Meter Drive, Amherst, MA 01002. (413) 549-6768.

THE MIND MACHINE is an educational program great for study and review. Quiz yourself on any facts for fast learning. Completely menu driven. Applesoft/48K. Includes documentation. $29.95 postpaid. Computer Enrichment Co., 9596 Paragon Rd., Centerville, OH 45459; (513) 885-3384.

DESPERATELY WANTED: 1978 Call -A.P.P.L.E. Annual. If you have a complete set of the 1978 issues of Call -A.P.P.L.E. and are willing to sell, PLEASE let me know your price. All offers will be considered. Thanks! Ken Sherwood, 117 N. 25th St., Reading, PA 19606, (215) 779-2074.

CASTLE WOLFENSTEIN IMPROVED! Introducing The Great Escape Utility. A powerful, new machine-language program which enables you to easily modify every feature of Wolfenstein to your own specifications. Eliminate the lengthy start-up screens, wall collide sounds and the time it takes to open the chests. Carry hundreds of bullets, grenades, the plans, etc. Start in any room, at any rank. Improve or handicap your marksmanship to any degree you choose. And much more. This unique utility received rave reviews at the recent Anaheim Applefest, and now it's selling like crazy. It's like owning a whole new Wolfenstein. Includes disk, five page handbook, strategy hints and a free map of the Castle. Only $15. ($4 for map alone). Send to MOXIE, 2049 Century Park East, Suite 5264, Los Angeles, CA 90067. Each copy individually numbered and guaranteed.

7 P.M.: REGULAR: PASCAL USERS GROUP: 1408 N.E. 36th St: Seattle: For info, contact Jim Adkins, 633-3182. The group considers all form and dialects of Pascal and normally meets the third Wednesday each month. Location subject to change.

ZAP IS THE BEST DISKETTE sector editing program around! It can undelete files, customize catalogs, rewrite track/sector lists, compare sectors, etc. Manual included. Unprotected diskette. Applesoft, DOS 3.3 required. Send $10.00 check to: Stephenson Software, Route 1 Box 296, Marshalltown, IA 50158.

PRINTER SUPPLIES AT LOW PRICES. Ribbons, paper and labels. For price list — write or call SUNSET COMPUTER SERVICES, P.O. Box 781-F, Wheeling, IL 60090; (312) 459-1030.

RENT SOFTWARE $5 MONTHLY. No membership fee. Send SASE to: Preowned Software, 4229 Baker Ave. NW, Seattle, WA 98107 for list.

```
10  REM

  SO WHAT DID YOU EXPECT
     DEPT

STOWE KELLER : CALL
     -A.P.P.L.E.


100  NEXT WITHOUT FOR
     ERROR
```

PRODUCTS NOW AVAILABLE TO A.P.P.L.E. MEMBERS

Call -A.P.P.L.E. Specials

HSMO	Mockingboard Special	$175[3]
STSS-HP28	THE Spreadsheet Memory Special	$325[2]

Back Issue Specials

PSCA10	Group of ten	$20.00
PSCA15	Group of 15	$24.00
PSCA20	Group of 20	$28.00
PSCA25	Group of 25	$32.00

No Phone Orders

A.P.P.L.E. LIBRARY

PAO1	"Apple Orchard" No. 1 (magazine)	$ 2.50
PAUG	Apple Users Guide	$13.00
PASL	Assembly Lines: The Book	$17.00
PBAD	"Beneath the Apple DOS"	$17.00
PCP1	"Best of Cider Press" (S.F. Newsletter Annual) Vol. 1	$ 4.50

Call -A.P.P.L.E. Annuals

PPK2	1979	$10.00
PPK3	1980	$15.00
PPK4	1981	$18.50
PPK5	1982	$18.50

Call -A.P.P.L.E. In Depth

PID1	1: All About Applesoft	$ 7.00
PID2	2: All About Pascal	$11.00
PID3	3: All About DOS	$13.50
PKID	Kids and the Apple	$16.00
PVIS, PVI2	Power of VisiCalc Vol. 1 or 2, ea.	$ 8.50
PU65	Using 6502 Assembly Language	$17.00
PWTA	Woz Talks (VHS Video cassette)	$27.50

AS IS SOFTWARE

SPDS	Public domain diskettes ea. (Send 37¢ S.A.S.E. for list of titles)	$ 4.00
SSEA	Eamon Adventure Diskettes ea.	$ 4.00

A.P.P.L.E. DISKPAKS

SA1, SA2, 3, 4, 5, 6, 7, 8, 9A, 9B	Call -A.P.P.L.E. Anthology Vols 1-9B incl. ea.	$12.50
SDIC	DICE 82[7]	$12.50
SD6B	Diskpak 6B-16 (DOS 3.3 disk utility)	$18.50
SD7	Diskpak 7	$17.50
SIN1	In Depth 1 Diskette	$16.50
SIN2	In Depth 2 Diskette Set	$20.00
SIN3	In Depth 3 Diskette Set	$15.00
SPA1	Pascal Anthology '81	$12.50

A.P.P.L.E. HARDWARE

HAPC	ApController	$85.00
HAKK	Ap Coolkit	$20.00
HAPD	Ap DRIVE	$240.00
HAPD, HAPC	Ap DRIVE with Controller	$325.00
HLC1	Ap L Case 1	$30.00
HLC2	Ap L Case 2 (Lower case chip)	$25.00
HDBA	Disk Banks	$ 4.50

AP• Jewelry

HPIN	Lapel pin	$ 3.00

A.P.P.L.E. PRESTIGE Game Diskettes

SCCO	Cosmic Combat	$12.50
SMCU	Mummy's Curse	$12.50

Apple Prestige Product LinE

SAPM	Ampermanager	$22.50
SAPF	APfile — Text file utility	$18.50
SFLR	ApGraphics: The Filler	$17.00
SAM3	Apmail] [Version 3.0	$30.00
SCAR	Applesoft Carpenter	$18.50
SAPT	Aptest	$18.50
SMLC	Big Mac Macro-Assembler/Ted	$28.50
SGPL	Global Program Line Editor	$38.50
SHFI	Higher Fonts 1	$13.00
SHTP	Higher Text Plus	$27.50
SPGE	Program Global Editor	$18.50
SPLE	Program Line Editor	$20.00
SS70	Soft Seventy	$20.00
SSYM	Symbol Symon	$20.00
STSS	THE Spreadsheet Version 2.0	$75.00
STSC	THE Spreadsheet 1 → 2 File Converter	$ 5.00
SWW3	WordWeaver ///	$75.00

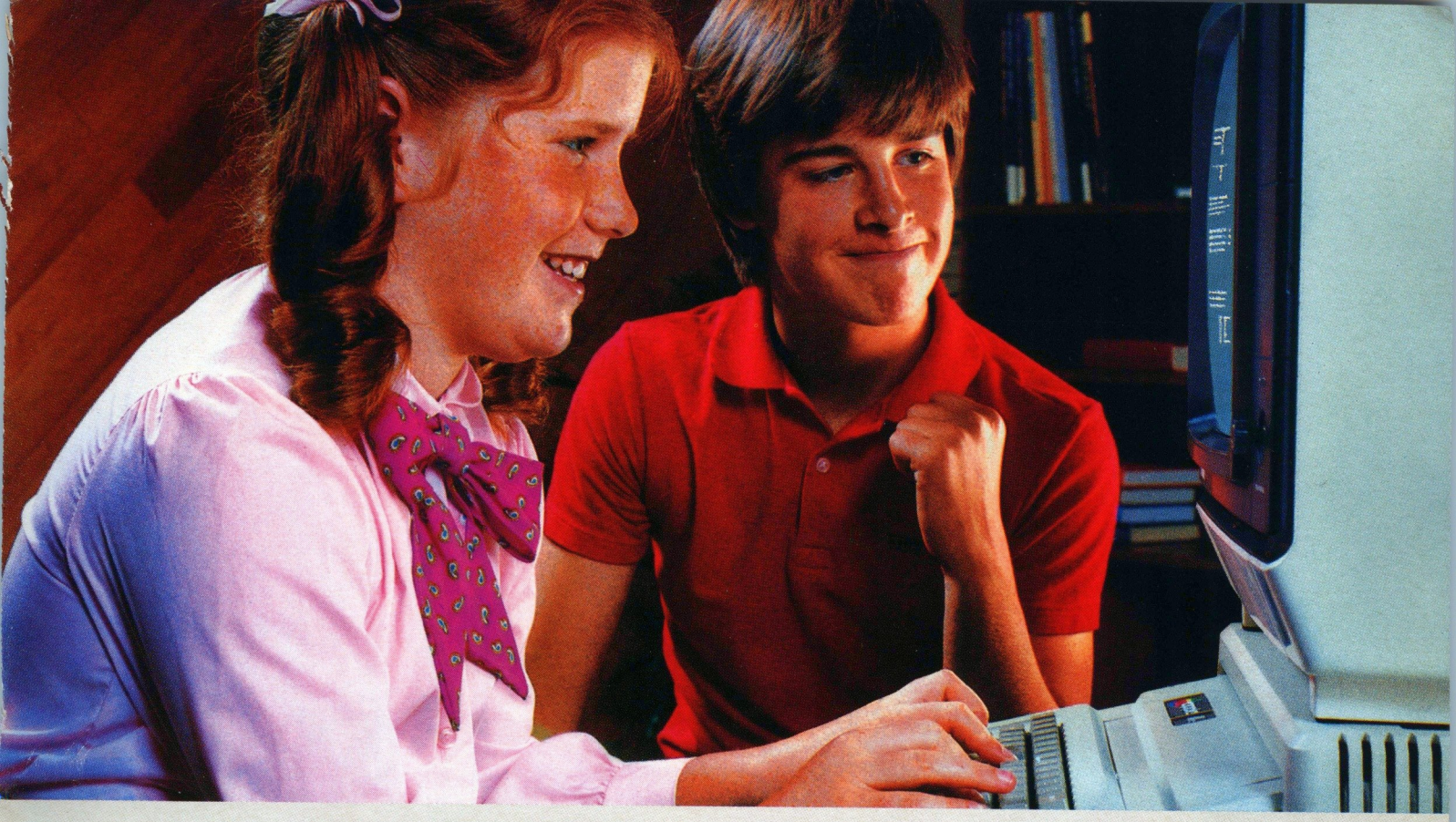
[2] Expires Nov. 1, 1983
[3] Expires Sept. 30, 1983
[7] $7.50 with Ap L Case 1
$10.00 with Ap L Case 2

Diskettes supplied on DOS 3.3 except as noted.

NOTICE TO FOREIGN MEMBERS:
A.P.P.L.E. Hardware products not available to overseas or foreign members.

Washington state residents add 7.9% state sales tax.

ORDER FORM ON REVERSE



PLATO COURSEWARE MAKES SERIOUS EDUCATION FUN.

Control Data PLATO® courseware brings new excitement to education. Through such action concepts as darts, pinball, hangman and highly interactive graphics and texts, PLATO challenges kids to reach their learning goals. PLATO coaxes and guides students. PLATO rewards them with the feeling of accomplishment.

PLATO courseware is being used in classrooms across the nation. Now you can bring this quality education home. You'll find PLATO lessons at selected retail outlets where quality software is sold.

All PLATO micro courseware is available for the Apple II Plus and Apple IIe. Lessons are also available for the TI 99/4A and the Atari 800.

There are lessons in **Math**, including Basic Number Facts, Whole Numbers, Decimals and Fractions; **Foreign Language** vocabulary builders in French, German and Spanish; a **Physics** lesson in elementary mechanics; **Computer Literacy** that introduces kids and novices to the computer age; ***Computer Concepts** that provide a springboard to more advanced computer lessons; and a ****Keyboarding** lesson that uses the famous Gregg typewriting approach to teach keyboarding techniques.

Widen your child's world. See the entire growing library of PLATO quality courseware at selected retail outlets. Or for information and a free PLATO catalog, call toll-free: 800-233-3784. Or write: Control Data Publishing Co., P.O. Box 261127, San Diego, CA 92126. In California call 800-233-3785.

Warranty available free from Control Data Publishing Co., 4455 Eastgate Mall, San Diego, CA 92121.

* Developed with Continuous Learning Corporation
** Developed with Gregg/McGraw-Hill.

Circle 13 on Reader Service Card

PLATO
COMPUTER-BASED EDUCATION

CONTROL DATA
PUBLISHING

BUY A BANANA.™ SAVE A BUNCH.

Meet the Banana,™ the very tough, versatile, portable, and reliable dot-matrix printer from Gorilla.™

At $249.95 retail it's an ideal and inexpensive companion for personal computers like Apple® (or Apple "look alikes" such as Franklin® or Albert®), TI,® Commodore,® TRS-80,® Kaypro,® Timex,® Osborne,® etc.

After that, it's merely comparable to other printers that can cost up to three times as much: 10″ carriages (to handle standard 9½″ paper), 80 columns, graphics capability, 10 characters per inch (expandable to 5 cpi) draft-quality print (for perfectly acceptable form letters, data processing, business reports, etc.) tractor feed (for precise alignment and quick loading), parallel or serial interface (take your pick), self-inking ribbon cassette (for long life and easy installation), 10 portable pounds in weight, and compatibility with so many of the most popular personal computers on the market.

Plus its printhead features a unique single-hammer design that eliminates a lot of moving parts, to eliminate a lot of annoying repairs.

That's the Banana: silly name, serious service. It's everything the expensive dot-matrix printers are . . .

Except expensive.

Leading Edge Products, Inc., 55 Providence Highway, Norwood, MA 02062.
Dealers: Call toll-free 1-800-343-8413; or in Massachusetts call (617) 769-8150.

Circle 29 on Reader Service Card